PERSONALNI

računari

93

JUN-AVGUST
1993.
CENA
4000000 DIN.
DEM 10 o USD 6 o ATS 70





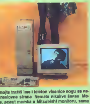
PERSONALNI
računari 93

časopis za informatiku i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
jun-avgust 1993

Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17
Generalni direktor
Ilija Rapajć
Glavni i odgovorni urednik
Jovan Regasek
Urednik
Ranka Jovanović
Komercijalni urednik
Vesna Jeremić
Tehnički urednik
Radoja Kavedžić
Stručna redakcija
Nenad Batocanin (baze podataka) Ranka Jovanović (aktuelnosti) Dejan Ristanović (softver) Jovan Skuljan (programiranje) Vladimir Slamenović (radne labele) Dejan Veselinović (hardver) Dragutin Vuković (mreže) Zoran Živolić (softver)
Adresa redakcija
11000 Beograd
Bulevar vojvode Mišića 17/III
Telefoni
647-476 (glavni urednik)
653-748 (redakcija)
651-666 (centrala, pretplata 226, stari brojevi 259)
Agencija BIGZ-a
(011) 651 793 653-565
Telefaks
(011) 648-140 647-955
Pretplata za zemlju
Za 3 meseca **zvati prodaju**
Za 6 meseci **zvati prodaju**
Za 11 meseci **zvati prodaju**
(na žiro-račun D P BIGZ, 60802-603-23264)
Pretplata za inostranstvo
Za jednu godinu 51 USD 80 DEM 67 CHS 27 GBP 271 FRF
(na devizni račun D P BIGZ 60811 620-16101 820701 999 03377)
Zbog usporenog prenošenja uplata preko banke molimo pretplatnike da nam posle svake nove uplate odmah pošalju foto kopiju uplatnice
Rukopisi se ne vraćaju

SEZAM (011) 648-422 (15 linija)
sistem za modemske komunikacije
Upravnik sistema
Jovan Regasek
Administratori na sistemu
Zoran Živolić i Dejan Ristanović

SADRŽAJ



Nemojte tražiti ime i telefon vlasnice nogu sa naše naslovne strane Nemate nikakve šanse Možete, poput momka u Mitsubishi monitoru, samo da – gledate Zato bolje okušajte sreću sa prvom nagradom u našoj nagradnoj igri – Info-D mašinom u klasičnom ISA kroju na cistih 50 MHz. Više detalja o ovom računaru, koji je i nas iznenadio svojim performansama, možete naci u tekstu „U apsolutnoj kategoriji" na stranama **14-16** a propozicije igre potražite na stranama **82-84**. Naslovna strana je, kao i obično, nastala u fotografskoj radionici *Studio Tešić* uz asistenciju g Nenada Petrovića

SADRŽAJ OGLASA

DETALJE O NAGRADNOJ IGRI POTRAŽITE NA STRANAMA 82-84

SLEDEĆI BROJ „RAČUNARA" IZLAZI IZ ŠTAMPE KRAJEM AVGUSTA

SOFTVER

Tržište

DOS 6.0 ruši rekorde

Samo u prvom mesecu prodaje nova verzija „Microsoft"-ovog najpopularnijeg proizvoda operativnog sistema *MS-DOS* u verziji 6.0 kupcima je isporučeno tri miliona paketa. Time je srušen raniji rekord u prodaji koji su zajedno držali *DOS 5.0* i *Windows 3.1*.

Pored toga *DOS 6.0* konkuriše i za titulu najkontraverznijeg proizvoda jer istovremeno sa više nego dobrom prodajom idu i učestali izveštaji o problemima u radu. Ugledni časopis „InfoWorld" je prikupio podatke o većini problema na koje su korisnici naišli i proverio ih u sopstvenoj laboratoriji za testiranje. Prema ovom izvoru ima dosta problema u radu sa diskovima koji koriste *DoubleSpace* za povećanje kapaciteta, a poruke poput „sector not found", „damaged CVF", „cross linked files" ili „lost cluster" su zabrinjavajuće česte. Većina ovih i sličnih problema iščezava kada se *DoubleSpace* ukloni.

Na sve ovo „Microsoft" odgovara da se radi o malom procentu prijavljenih problema u radu s obzirom na broj prodatih primeraka. Ipak, izgleda da *DoubleSpace* počinje da zadaje prilično glavobolje Bilu Gejtsu i kompaniji. Sa jedne strane, to su problemi u radu ovog programa, a sa druge, po svemu sudeći nova velika kompjutarsko-sudska afera oko patentnih prava. Da li *DOS 6.0* pati od bubica ili ne, uskoro će se veoma pouzdano znati, ali je indikativno da se izvestan broj proizvođača računara odlučio da malo sačeka sa isporukom nove verzije *DOS*-a sa svojim mašinama.

Operativni sistemi

Windows NT: Krajem jula po promotivnoj ceni



Višemesečna nagađanja konačno su se pretvorila u stvarnost. *Microsoft* je zvanično promovisao *Windows NT*! Za razliku od MS-DOS-a 6.0 koji je promovisan na (za tu priliku specijalno sazvanom) sastanku jednog korisničkog kluba, *Bill Gates* je odlučio da prezentira svoj novi operativni sistem u okviru uvodnog predavanja na poznatom sajmu *Comdex* 25. maja ove godine. Događaj je, baš kao i kada se radilo o DOS-u 6.0, prenosilo više lokalnih i satelitskih televizija, tako da su Gejtsov govor čule stotine hiljada ljudi na svim kontinentima.

Pokazalo se, na žalost, da promocija ne označava i početak isporuke – prvi primerci *Windows NT*-a isporučivaće se krajnjim korisnicima tek poslednjih dana jula. Oni koji planiraju da koriste NT na mrežnim serverima, moraće da se strpe 30 dana više, dok će korisnici DEC-ovih *Alpha* stanica (NT je zamišljen kao operativni sistem koji nije vezan za jedan računar, pa prema tome ni za jedan procesor) sačekati daljih 30-60 dana.

To su bile loše vesti sa promocije, a bilo je i dobrih, naročito kada se radi o cenama. Vlasnici neke od verzija *Windows*-a ili OS/2 mogu da dobiju *Windows NT* za manje od 300 dolara, što je znatno niža cena od ranije pominjanih 500 ili čak 900 dolara. Naravno, oni koji pretenduju na *Advanced Server Edition* moraće dublje da posegnu u džep i plate bar 1495 dolara, što je zapravo promociona cena – posle prvih šest meseci NT će se prodavati po punoj ceni od oko 3000 dolara.

Gejts je između ostalog rekao novinarima da će NT biti prvi operativni sistem koji kombinuje *client-server* aplikacije sa glavnim poslovnim aplikacijama. „To će", kaže Gejts, „biti ništa manje nego fundamentalna promena načina na koji kompanije formulišu svoje poslovno-računarske zahteve. Sav biznis koji se oslanja na računare kada će moći da se obavlja na istoj platformi, u punoj meri uživajući blagodeti *client server* rešenja i poslovnih programa. Sada će projektanti sistema moći da standardizuju rad, obezbede visoke performanse i integrišu sve aplikacije koje se u njihovim centrima koriste. Biće im lako da sakupe sve podatke kojima korporacije raspolažu na isto mesto i tako znatno brže odgovore na zahteve onih kojima je analiza tih podataka najpotrebnija."

Zanimljiva i neočekivana novost je i to što će *Microsoft* staviti javnosti na raspolaganje kompletan izvorni kod (*source*) NT-a – na taj će način biti olakšan razvoj aplikacija, a fakulteti i druge naučne institucije biće u prilici da detaljno prouče čitav sistem i eventualno predlože njegova unapređenja.

Ako se po jutru dan poznaje, budućnost *Windows NT*-a je prilično svetla – pojavilo se veliko interesovanje nezavisnih proizvođača softvera i trenutno je u razvoju preko 75 novih 32-bitnih komercijalnih programa koji će raditi pod NT-om, očekuje se da ih do kraja godine bude preko 250. Smatra se da će nekoliko „velikih" NT programa biti spremno za isporuku već u avgustu i septembru.

D.R.

Komunikacije

Bežične mreže: Čeo sam na kompjuteru...

PCS (*Personal Communications Services* – personalne komunikacione usluge) ideja o javnoj bežičnoj mreži u SAD dobija sve više podrške. Za razliku od informativne magistrale kontroverznog plana predsednika Klintona, koja bi povezivala velike kompanije i univerzitete, PCS mreža bi bila otvorena i za obične amerikance. Ako PCS koncept bude prihvaćen, veliki broj radio-frekvencija koje drži američka vlada postaće dostupan javnosti.

PCS se sastoji iz nekoliko različitih komponenti – neke su besplatne, a neke ne. Besplatan je *User PCS* – kao i kod AM/FM radija, ukoliko imate ispravan *User-PCS* predajnik i prijemnik, radio frekvencije su besplatne, ali domet iznosi samo oko 50 m. Bežične LAN, PDA i PBX mreže i telefoni će podržavati *User PCS*.

Carrier-PCS će pokrivati širi prostor, ali neće biti besplatan. Predviđeno je da njima upravljaju poznate PTT kompanije (npr. AT&T i MCI) koje će od federalne vlade otkupiti prava na frekvencije, a zatim ih prodavati korisnicima. *Carrier PCS* se može uporediti sa drugim uslugama koje se plaćaju – pejdžeri, radio-pošta, bežične LAN mreže.

PCS može izgledati kao još jedan san o bežičnom prenosu informacija, ali je činjenica da iza njega stoji preko trideset kompanija koje su zaista neko i nešto u svetu kompjutera. Na čelu sa kompanijama "Apple" i "Hewlett Packard" ovih 30 velikih je stvorilo WinForum (*Wireless Information Networks Forum* – Forum za bežične informacione mreže) koji treba da predloži standarde Federalnoj komisiji za komunikacije. WINForum je tražio talasni opseg od 40MHz, ali Federalna komisija namerava da odobri samo 20 MHz.

No, sada će i da li će PCS uopšte biti prihvaćen veliko je pitanje. U Americi je u pripremi zakon o novoj telekomunikacionoj tehnologiji (*Emerging Telecommunications Technology Act*) koji treba da dozvoli upotrebu radio talasa, ali ga Senat još nije odobrio. Čak i ako se to desi, PCS telefoni i lokalne mreže se ne mogu očekivati pre 1995.

Programski jezici

MS Cobol odlazi u penziju

Prema poslednjim informacijama iz „Microsoft"-a izgleda da je došlo vreme kada ćemo reći „*COBOL* je deo računarske istorije!" Najveća softverska kuća na svetu odlučila je da od 30. juna prestane sa prodajom *COBOL* programskog jezika.

Stono izdavaštvo

Može i za malo para: PagePlus 2.0

Nedavno je predstavljena nova verzija programa za stono izdavaštvo *PagePlus 2.0*. Ovog puta program je napisan za *Windows* platformu i kompanija „Serif", proizvođač programa, očekuje da će biti ponovljen uspeh postignut sa verzijom 1.0 koja je najbolje prodavan DTP program u Engleskoj, a u ostalim evropskim zemljama je uvek pri vrhu popularnosti. Jedan od razloga za to je cena, značajno niže nego kod sličnih programa. Tradicija se nastavlja za *PagePlus 2.0*, za koji proizvođač kaže da će koštati samo 59.95 dolara.

Naravno, cena nije jedini kvalitet *PagePlus*-a. Dobro osmišljeni korisnički interfejs koji je omogućavao lak rad u verziji za DOS, pod *Windows*-ima je dobio i mogućnosti rada sa okvirima u koje se smeštaju tekst i grafika. *PagePlus 2.0* je obogaćen i radom sa bojama, predefinisanim stilovima strana, rotacijom teksta i objekata, kerningom, podrškom za OLE i konvertorom formata datoteke za *Windows* tekst-procesore.

Kompanija „Serif" pripremila je i pakete dopunskih programa za *PagePlus 2.0*. *TypePlus* omogućava obradu teksta, a *FontPack* vašu kolekciju pisama dopunjuje sa još 100 *TrueType* slovnih likova.

Kontakt adresa: Serif inc., tel 603-889-8650

Obrada slika

Novi Hijaak za Windows: Napisano iznova

Jedan od najpopularnijih programa za konverziju grafičkih formata i „hvatanje slike" sa ekrana monitora počeo je da se prodaje u novoj verziji *Hijaak Pro 2.0 for Windows*. U odnosu na verziju 1.0 za *Windows*-e povećana je brzina rada, poboljšan je korisnički interfejs, dodata je podrška za nove grafičke formate i za *TWAIN*.

Proizvođač ovog programa, „Inset Systems", tvrdi da je veliki deo koda ponovo napisan sa namerom da se poboljšaju performanse, naročito kod otvaranja i učitavanja datoteka. Pored obilja formata koje je i ranije *Hijaak* poznavao, dodati su još neki. Tako je sada moguće, na primer, direktno sa CD-a učitati slike zabeležene u *Kodak Photo compact disc* formatu. Podržana je i *Aldus Graphic Import Filter* specifikacija.

Kao opcija programa dodata je i *JPEG* (*Joint Photographic Experts Group*) kompresija. Tako se sa *Hijaak 2.0 for Windows* mogu pregledati, čitati, zapisivati *JPEG* datoteke sa stepenom kompresije do 200 prema 1. Nisu zaboravljeni ni korisnici kojima je potreban prenos grafičkih fajlova sa drugih računara. Podržani su „Sun Unix" radne stanice, „IBM"-ovi mejnfrejm računari i „IBM AS/400" koji koriste *IOCA* format.

U kompaniji „Inset" posebno ističu da *Hijaak* sada ima mogućnost da radi sa dokumentima koji su zapisani prema *TWAIN* specifikaciji. Iza ove skraćenice krije se standard podržan na velikom broju računarskih platformi, tako da obezbeđuje kompatibilnost među aplikacijama i ulaznim aplikacijama kao što su grafičke baze podataka ili skeneri. *Hijaak 2.0* direktno uvozi sliku iz jednog od mogućih *TWAIN* „izvora" podataka i sliku beleži u bilo kom rasterskom formatu koji je podržan.

Hijaak Pro 2.0 se prodaje u Americi po preporučenoj ceni od 169 dolara, što znači da se u radnjama može i jeftinije kupiti. Registrovani korisnici prve verzije *Hijaak*-a za *Windows*-e plaćaju samo poštanske troškove.

Kontakt adresa: Inset Systems, tel. 203-678-0012, fax 203-677-6513

Obrada slika

Ventura PicturePro – još jedna alternativa za obradu slika

„Ventura Software" stupa na prilično popunjeno tržište softvera za obradu slika sa izuzetno opremljenim profesionalnim paketom, koji objedinjuje izvanredne alatke za maske i kompoziciju, fleksibilne alatke za bojenje kao i dobar izbor specijalnih efekata. Iako *PicturePro*, verzija 1.1, u mnogo čemu podražava mogućnosti *Photoshopa* (vidi opširniji prikaz u ovom broju), kao i programa *Fractal Design Painter*, to čini unutar svog dobro prepoznatljivog interfejsa.

Rad se obavlja u jednom od tri režima: *Paint*, *Alpha* ili *Draw*. U režimu za crtanje *Draw* kreiraju se objekti u geometrijskim oblicima i slobodne Bezier krive. Kao u svakom paketu za obradu ilustracija, objekti se mogu ređati jedan povrh drugog (stack) i premeštati. Ako objekat popunite bojom ili teksturom, on se kopira u sliku kao bit-mapa, a bez teksture postaje prostor za selekciju (*selection area*). Ventura VIP format posebno čuva *Draw* sloj (kao i *Alpha* sloj), ostavljajući mogućnost da se edituje sve dok ne odlučite da ga prenesete na sliku koja je u pozadini. EPS datoteke se mogu uvoziti, editovati Bezier alatkama i koristiti kao maske.

Režim *Draw* čini program *PicturePro* posebno fleksibilnim, i to na dva važna načina. Kao prvo, tekst unesen u *Draw* sloj ostaje otvoren za editovanje sve dok se ne prenese na sliku. Kao drugo, u ovom režimu na raspolaganju su moćne alatke za editovanje Bezier krivih u raznim varijantama: glatke simetrične, asimetrične, linija, kriva, isečak kruga. Osim toga, tekst ili objekat se može pretvoriti u Bezier krivu.

Celokupno editovanje i bojenje slika se obavlja u režimu za bojenje, *Paint*. *PicturePro* nudi uobičajeni široki spektar filtera za specijalne efekte (predviđeno je i priključivanje filtera drugih proizvođača), alatke za korekciju boja i transformacije (iskošenje, rotacija, perspektiva). Slika se može konvertovati u CMYK model boja, ali se tonska mapa i alatke za korekciju mogu koristiti samo za CMYK editovanje.

U *Alpha* režimu svi selektovani elementi se tretiraju kao 8-bitne maske, na koje se mogu primeniti sve alatke za editovanje. Ove alatke, od kojih neke imaju neobične nazive (npr. potez kistom može biti poput „meke četke za uljane boje" ili „istrošenog flomastera") neodoljivo podsećaju na one iz programa *Fractal Design Painter* – doduše, malo su manje realistične, ali su još uvek izvrsne. Nude kontrolu nad bezbroj različitih poteza kistom, kao i interakciju između poteza i *Pixel/Paper* podloge.

PicturePro ima drugačiji pristup kompoziciji nego *Photoshop*, ali takođe postiže izvanredne efekte kombinovanjem dve slike koristeći transparentnost treće. Program podržava drajvere za skenere i *Twain* drajver, i ima izuzetne mogućnosti separacije boja.

Sve u svemu, *PicturePro* ima izvanredno praktičan i dosledan interfejs. Ctrl-klik daje opcije za bilo koju alatku iz palete, dok klik na polja drugih paleta daje većinu opcija iz *drag-and-drop* menija.

Iako se Ctrl-klik princip u praksi pokazuje prilično dobro, ipak zahteva upotrebu obe ruke, što ponekad može biti nespretno. Osim toga, neki elementi koji se dosta često koriste nalaze se barem za jedan nivo niže nego što bi trebalo.

Nedostatak je i to što program nema sposobnost konvertovanja u 8-bitne slike niti editovanja ovakvih slika. Kao alatke za foto-retuširanje *PicturePro* se može meriti sa većinom sličnih paketa, ali ne i sa *Photoshop*-om. Alatke za maske su izvrsne, ali bi lepo bilo da je i maskiranje polja boje rešeno – kao kod programa *Picture Publisher* (Micrografx).

Što se performansi tiče, potrebno je dosta podešavanja, jer čak i kod rada sa relativno malim slikama (768K) i dvojno slobodne virtuelne memorije (13MB) može doći do zastoja mašine.

Posmatrano pojedinačno, sposobnosti paketa *Ventura PicturePro* se ničim ne ističu u odnosu na svoje rivale. Međutim, kada se uzmu u obzir sve zajedno – izvanredno editovanje Bezier krivih, rad sa maskama i alatke za bojenje – onda se *PicturePro* pokazuje kao moćno sredstvo za profesionalno kreiranje i obradu slika.

Cena – 795 dolara

Zahtevi 4MB RAM, tvrdi disk od 10MB, *Microsoft Windows 3.x*, miš; preporučuje se 8MB RAM, digitizing tablet i matematički koprocesor

Kontakt adresa: Ventura Software Inc., 15175 Innovation Dr., San Diego, CA 92128, tel. 800-822-8221, 619-673-0172, fax 619-673-7784.

Obrada slika

Nova tehnologija za obradu bit-mapa

Koristeći tehnologiju staru gotovo koliko i televizija, "Altamira Software Corp." i "Micrografx" će učiniti da PC napravi ogroman korak napred u oblasti kreiranja i obrade bit-mapiranih slika. Naime, obe kompanije predstavljaju nove proizvode za lakše editovanje slika.

Kod ovog novog pristupa svaka bit-mapa u okviru složene slike se tretira kao objektni sloj. To znači da možete kombinovati, na primer, jato ptica sa slikom oblačnog neba, a da se pri tom ne izgubi mogućnost pomeranja jata. Može se postići i realističan efekat jata koje prolazi kroz oblake. Kao i kod vektorskih programa za obradu ilustracija, slojeve možete poređati bilo kojim redosledom i premeštati ih bezbroj puta.

Sa današnjom rasterskom tehnologijom, kada postavite ptice na određeno mesto, one postaju deo jedne mape piksela, a njihovo premeštanje je mukotrpan proces, jer treba precizno odabrati prostor masku, isecati, lepiti i uklapati. Pošto jato dolazi sa ugrađenom maskom, ono je unapred selektovano i nema pozadine kao što bi bilo uobičajeno.

"Altamira" svoj proizvod naziva *Dynamic Alpha*, a prvi put će se pojaviti u sklopu paketa za editovanje slika, *Altamira Composer*, koji se očekuje uskoro. Svaki objektni sloj će imati svoj alfa-kanal (transparentna ravan koja se u programima za obradu slika obično naziva maskom). Svaki element je zasebni sloj koji se može selektovati sve dok ne odlučite da ga prenesete u EPS ili TIFF datoteku. U "Altamiri" se nadaju da će obogatiti biblioteke slika (u njihovoj terminologiji "crop art") proširujući standardne TIFF, JPEG, Targa i BMP datoteke, koje obezbeđuju podršku alfa kanala. "Altamira" namerava i da prodaje licencu za korišćenje nove tehnologije, tako da ćemo je možda videti i u drugim programima za obradu bit-mapa.

Micrografx je razvio sopstvenu shemu za raspored slojeva, koja će biti ugrađena u novu verziju paketa *Picture Publisher*. Da bi čuvao memoriju, *Micrografx* ne zauzima poseban kanala za svaki sloj, već samo jedan alfa-kanal koji zatim deli slojevima.

Grafički softver

Čekajući 4.0: Manje para za CorelDraw 3.0

Ako ste nameravali da kupite *CorelDraw 3.0*, možda je sada pravo vreme da to učinite. Proizvođač ovog izuzetno popularnog programa je spustio cenu za čitavih 400 dolara. Tako se sada *CorelDraw 3.0* može kupiti za samo 199, odnosno 249 dolara, u zavisnosti od toga koju verziju paketa kupujete.

Razlog za pojeftinjenje je priprema terena za novu verziju programa. *CorelDraw 4.0* prodavaće se paralelno sa verzijom 3.0, a nudiće mogućnosti koje bi trebalo da zadovolje i najzahtevnije korisnike. Tako će aktuelna verzija postati ulaznica u *CorelDraw* svet, dok će „četvorka" biti namenjena profesionalcima.

Kontakt adresa: Corel, tel. 613-728-8200 ext 1672

Obrada teksta

Jednostavan tekst procesor za Windows

„Symantec" je izbacio *Windows* verziju svog tekst procesora *Q&A Write* za koju navodi da štedi prostor na disku i memoriju računara, a da pritom ne narušava komfor u radu.

Proučavajući zahteve korisnika koji svakodnevno koriste neki od popularnih tekst procesora, „Symantec" je izolovao najvažnije mogućnosti koje treba da ima ovakav programski paket i ugradio ih u *Q&A Write 3.0 for Windows*. To su provera pravopisa, tezaurus, mogućnost ugradnje grafike, teksta i tabela, direktni „mail merge" iz najpopularnijih baza podataka, fax komunikacija, elektronska pošta i transparentna konverzija tekstova pisanih u *Microsoft Word*-u i *WordPerfect*-u.

U želji da svoj tekst procesor za *Windows*-e učini što manjim, „Symantec" je sve, po njima, nepotrebne opcije izbacio tako da *Q&A Write 3.0* na disku zauzima od 1.7 MB (minimalna konfiguracija) do 4 MB za punu instalaciju. Koliko je to malo za jedan *Windows* tekst procesor najbolje govori poređenje sa *Microsoft Word*-om koji zauzima 15 MB.

Q&A Write 3.0 for Windows je samo prethodnica drugom, još poznatijem proizvodu „Symantec"-a – bazi podataka *Q&A* koja bi uskoro trebalo da se pojavi u verziji za *Windows*-e. Termin niko u „Symantec"-u nije spreman da jasnije definiše, što znači da proizvod ne treba očekivati tako brzo. Ono što se u ovom trenutku sasvim pouzdano zna je cena: *Q&A Write for Windows* košta 69.95 dolara.

Obrada teksta

MS Word 6.0: Trka se nastavlja

Bez obzira što su *Windows* programi zavladali na tržištu, dva najpopularnija programa za obradu teksta pod DOS-om, *WordPerfect* i *MS Word*, nastavljaju nesmanjenom željom da se nadmeću za primat. Tako je početkom maja, na velika zvona predstavljen *Microsoft Word for DOS* u verziji 6.0. Recimo i to da je promocija obavljena simultano u Australiji, Kanadi, Nemačkoj, Velikoj Britaniji i Americi.

Podsetimo se malo (bliže) prošlosti ovog tekst procesora. Verzija 5.0, veoma popularna kod nas, predstavljena je novembra 1988. godine. Dva leta kasnije pojavila se verzija 5.5, koja je donela padajuće menije i ostale „novotarije" tog vremena. Dinamika je nastavljena i sada je pred korisnicima programa za obradu teksta sa 286 i slabijim 386 računarima *MS Word 6.0*.

Od mogućnosti koje donosi novi *Word* za DOS pominjemo „drag-and-drop" editovanje, opciju za formiranje tabela, podršku *TrueType* formata slovnih likova. Tu je još i režim za pregled strana pre štampanja u visokoj rezoluciji, a struktura menija je usaglašena sa drugim „Microsoft"-ovim programima, poput *Excel*-a ili *PowerPoint*-a.

Uz *Word for DOS 6.0* isporučuje se i program *Grammatik*, namenjen proveri gramatičke ispravnosti teksta pisanog, naravno, na engleskom jeziku. Korisnicima ranijih verzija *Word*-a namenjena je opcija koja ih upoznaje sa novinama i pomaže lakšem prelasku sa stare na novu verziju. Onima koji prelaze sa najvećeg konkurenta, programa *WordPerfect*, namenjen je poseban vodič. Pomenimo još i konverziju datoteka. Podržani su svi značajniji formati, tako da prelazak sa nekog drugog tekst procesora ne znači gubitak mogućnosti da se i dalje manipuliše sa starim tekstovima.

Da biste pokrenuli *Microsoft Word 6.0* potreban vam je PC računar sa najmanje 384 kB RAM-a (preporučuje se barem 512 kB) i više od 1.5 MB prostora na tvrdom disku. Puna instalacija programskog paketa zahteva 5.5 megabajta. Preporučena cena je 495 dolara, a prelazak sa stare verzije košta 99 dolara za one koji to urade do septembra.

Kontakt adresa: Microsoft Corporation, tel. 206-882-8080

Radne tabele

Lotus 1-2-3: Pod Windowsima četvrti put

Najpoznatiji program za unakrsna izračunavanja, *Lotus 1-2-3* ne želi tako olako da se preda pred konkurencijom na tržištu ovih progama koji rade pod *Windows*-ima. Najavljena je nova verzija *1-2-3 for Windows*, sa oznakom revizija 4.0. Znači, ovog puta se spremaju veće promene programa.

Prema navodima kompanije, prvi put će biti primenjen *Lotus Chronicle version-control* tehnologija, a mesto će naći i kontekstno osetljiv korisnički interfejs, sve u cilju da se učenje i rad sa programom učine lakšim. Sve u svemu, čini se da „Lotus" pokušava da uhvati korak sa programima *Excel 4.0* i *Quatro Pro for Windows* koji se trenutno bore za primat na tržištu *Windows* verzija radnih tabela.

Lotus 1-2-3 Release 4 for Windows treba da se pojavi na američkom, japanskom, francuskom i nemačkom tržištu krajem juna, dok se verzije namenjene drugim jezičkim područjima očekuju u trećem kvartalu ove godine.

Kontakt adresa: Lotus Development Corp., tel. 617 693-1283.


Unlock Your Mind
RAČUNARSKI SISTEMI
LICENCNI SOFTVERI
KOMPONENTE
ŠTAMPAČI

ŠTA IMA NOVO

Softverske alatke

Alat za diskove pod DoubleSpace-om

Kompanija „Addstor" je napravila program koji omogućava održavanje komprimovanih datoteka i diskova kreiranih jednom od udarnih novosti DOS-a 6.0 – programom za kompresiju podataka *DoubleSpace*.

Program *DoubleTools* omogućava korisnicima da pregledaju, održavaju i proširuju *DoubleSpace*-om komprimovane datoteke. Program poseduje mogućnost da prikaže informacije o sistemu koje vode ka maksimizaciji kapaciteta i performansi. Proizvođač takođe tvrdi da nisu podržani samo tvrdi diskovi, već i flopiji, kao i prenosni mediji, na primer *Bernoulli* drajvovi.

Na promociji *DoubleTools*-a, predsednik „Addstor"-a Richard Preston izjavio je da njihov program omogućava da se sa *DoubleSpace*-om dobije bolji stepen kompresije. Ili ako vam je tako draže, povećavanja kapaciteta diska nego što pružaju konkurentski proizvodi *SuperStore* (DR DOS 6.0) i kod nas veoma popularni *Stacker*.

U okviru *DoubleTools*-a su moduli – *The Advisor* koji daje preporuke za optimizovanje performansi diska, defragmentator i alat za verifikaciju podataka koji radi u pozadini; *File Director* – set funkcija za premeštanje i kopiranje datoteka između kompresovanih i nekompresovanih flopi diskova, odnosno drugih prenosnih medija i *Universal Data Exchange (UDE)* koji omogućava da se kompresija obavi na prenosivim medijima. Interesantno je da *UDE* omogućava čitanje i zapisivanje na komprimovanim diskovima na svakom DOS sistemu, bez obzira da li je na računaru prisutan *DoubleSpace* ili *DoubleTools*.

Korisnicima će dobro doći i mogućnost ovog programa da pre kopiranja predvidi koliko će prostora zauzeti komprimovana datoteka, ili više njih. Ništa manje nije interesantna opcija koja, u slučaju da je grupa datoteka komprimovana i pohranjena na više disketa, podatke o mestu gde se nalaze sve datoteke postavlja na prvu disketu, tako da je lako utvrditi gde se tačno nalazi fajl koji nam je potreban.

Od opcija koje nudi *DoubleTools* pominjemo i *File Viewer* alatku za pregled sadržaja datoteke bez startovanja aplikacije u kojoj su kreirane. I na kraju, tu je i alat za oporavak oštećenih komprimovanih datoteka. Cena programa je 89.95 dolara.

Kontakt adresa: Addstor, tel. 415-688-0470

Baze podataka

SQL i Windows NT: Preliminarni razvojni paket

„Microsoft" je počeo sa isporukama SQL razvojnog paketa koji radi pod novim *Windows NT* operativnim sistemom. Prema podacima koje je prezentirala sama kompanija, radi se o preliminarnom razvojnom paketu, namenjenom softverskim kompanijama i velikim korisnicima čiji računari rade pod *Windows*-om *NT*.

Ovaj SQL paket omogućava razvoj klijent-server aplikacija koje mogu u potpunosti da iskoriste prednosti novog tridesetdvobitnog operativnog sistema. Prema rečima doktora Čarlsa Elmasa, rukovodioca ovog projekta, „Microsoft"-ov paket podržava *SQL server* standard u verziji 4.2, što znači da je klijent server arhitektura aplikacija u potpunosti inkorporirana.

Cena paketa je za povlašćene korisnike u Engleskoj 495 funti i radi pod beta verzijom *Windows*-a *NT* koja je objavljena u martu ove godine. Hardverski zahtevi su: procesor 386 ili jači, najmanje 12 MB RAM-a i 30 MB prostora na hard disku.

Kontakt adresa: Microsoft, tel. 0734-270001

Baze podataka

I dBFast u trci: Pomoću miša, od oka...

Poznati proizvođač softvera „Computer Associates International" je na tržište izbacio novu verziju svog programa za upravljanje i razvoj baze podataka *dBFast 2.0* radi pod *Windows*-ima i kompatibilan je sa Xbase standardom koji je promovisao *dBase IV*.

Kao i kod ostalih najnovijih programa za rad sa bazama podataka, u velikoj meri su iskorišćene prednosti *Windows* okruženja. Umesto često mukotrpnog kodiranja u nekom od editora, sada se dobar deo aplikacije u *dBFast*-u kreira vizuelno, pomoću miša i grafički prikazanih alata.

dBFast 2.0 u sebi sadrži i *CA RET* generator izveštaja, dibager i programerski editor koji istovremeno može da radi sa najviše osam dokumenata. Preporučena cena za ovaj paket je 550 dolara, ali će se u početku prodavati po reklamnoj ceni od 249 dolara, dok će korisnici ranijih verzija moći da pređu na novu za samo 99 dolara.

Kontakt adresa: Computer Associates, tel. 516-342-2391, fax 516-342-5329.

Baze podataka

Razvojni paketi za FoxPro 2.5

„Microsoft" je nedavno na tržište izvacio tri razvojna paketa za popularnu bazu podataka *FoxPro 2.5*. Prva dva imaju isto ime *FoxPro Distribution Kit*, a razlikuju se po tome što je jedan namenjen verziji koja radi pod DOS-om, a drugi za *Windows* verziju *FoksPro*-a. Treći paket nosi naziv *Library Construction Kit*.

FoxPro Distribution Kit omogućava programerima da aplikacije pisane u *FoxPro*-u kompajliraju i isporučuju kao zasebne programe. U okviru ovog paketa nalazi se i *Microsoft FoxPro Setup Wizard* alat za pripremanje instalacionih programa. Pored ovoga, verzija namenjena za rad pod *Windows*-ima uključuje i alat za pravljenje helpova u maniru ovog radnog okruženja. Takođe, tu je i „runtime" verzija *Microsoft Graph*, programa za pripremu grafika.

Drugi paket, *Library Construction Kit* namenjen je povećanju mogućnosti *FoxPro* aplikacija kroz povezivanje sa C-om. Podržani su *Microsoft C/C++ 7.0*, *Visual C++* kao i *Watcom C compiler 8.5*. Ostale mogućnosti koje donosi *Library Construction Kit* su kreiranje posebnih *FoxPro for Windows* biblioteka (.PLB) i nove funkcije za editovanje koje uključuju i rad sa tastaturom i mišem na niskom nivou.

Kontakt adresa: Microsoft GmbH, Edison strasse 1, 8044 Unterschleissheim, tel. 089-3176-38-38, fax 089-3176-38-30

Mreže

Novell i Windows NT: Novell nije sujetan

„Novell"-ov mrežni softver, prema nekim procenama, drži oko 70 procenata lokalnih mreža u svetu. Zato nimalo ne čudi što je ova kompanija nedavno predstavila beta-verziju *NetWare* klijentskog softvera, koji radi pod *Windows NT* operativnim sistemom. Time se nastavlja strategija „Novell"-a da se pokrivaju svi značajni operativni sistemi. Dosad su to bili DOS, OS/2, Unix i Macintosh.

Koristeći ovaj softver, računar pod *Windows NT*-om može pristupiti *NetWare* mreži kao klijent. Pri tom su dostupni IPX, SPX II i *NetWare*-ovi transportni stekovi. Naročito je interesantan SPX II koji, prema kazivanju „Novell"-ovih stručnjaka, poboljšava performanse i pojednostavljuje razvoj aplikacija koje rade u mrežnom okruženju.

Predstavljena beta-verzija nudi sledeće mogućnosti: *NetWare 4.0 ODI LAN* drajvere, print servis, integrisani program za instalaciju, kao i opciju koja omogućava da se koristi isto korisničko ime i lozinka za pristup kako *NetWare* mreži, tako i samom *NT*-u.

U sledećoj fazi razvoja ovog softvera predviđa se ugradnja *NetWare 4.0* servisa za rad sa diskovima, *NetWare Client API*, DOS i OS/2 emulacija. Za ljubitelje grafičkog korisničkog interfejsa interesantna je informacija da će „Novell" razviti odgovarajući tridesetdvobitni sistem za razmenu informacija između korisnika i računara, naravno u *Windows* stilu.

Mreže

Microsoft Mail 3.2: Jednostavniji rad

Nova verzija popularnog paketa za elektronsku poštu na mrežama *Microsoft Mail 3.2* donosi nekoliko unapređenja, pre svega u pogledu postavljanja i povezivanja sa mrežnim softverom, podrška X.25 protokolu i mogućnost da se korisnici lako mogu premeštati sa jedne „elektronske pošte" na drugu.



Mail 3.2 ima opciju automatskog otračuna za korisnike koji se kreiraju na osnovu postojećih mrežnih direktorijuma. Ako je instalirana *Novel NetWare* mreža, ovaj posao obavlja pomoćni program *Novext.EXE*. Time je znatno olakšan posao administratorima mreže, koji su dosad sve morali ručno da rade.

Kako mnogi veliki korisnici pored neke od LAN mreža poseduju i X.25 mreže, odnosno priključke na njih, „Microsoft" je u *Mail 3.2* uključio podršku i za ovaj protokol, tako da je omogućena razmena elektronske pošte i na ovaj način.

Poslednja značajna izmena u odnosu na prethodne verzije *Mail*-a odnosi se na olakšavanje rada administratora mreže. Premeštanje korisnika sa jednog na drugo mesto sada se obavlja na način koji je najlakše uporediti sa *drag-and-drop* funkcijom u *Windows*-ima.

Kontakt adresa: Microsoft GmbH, Edison strasse 1, 8044 Unterschleissheim, tel. 089-3176-38-38, fax. 089-3176-38-30.

Mreže

MS Office: Sve u jednom paketu

U poslednje vreme „Microsoft" pokazuje znake pojačane aktivnosti na tržištu. Tako je posle DOS-a 6.0, *Visual C++* paketa i drugih novosti najavljena i revizija paketa *Microsoft Office*.

Verzija *Office*-a koja se dosad nalazila, i još uvek se nalazi na tržištu, okupljala je na jednom mestu *MS Excel 4.0*, *MS Powerpoint 3.0*, *MS Word* i licencu za *Microsoft Mail*. U novoj verziji će ovom skupu biti pridodat i novi paket za rad sa bazama podataka *Microsoft Access*.

Prema navodima kompanije, i *Access*, koji je od novembra do maja prodat u 800 hiljada kopija, i *Office* se uglavnom nalaze u vrhu prodaje među sličnim paketima, a zvaničnici „Microsoft"-a ističu i to da je prošle godine više od polovine paketa *Excel*-a i *Word*-a isporučeno upravo u okviru *Office*-a. Stoga je prirodno da se i *Access* uključi u „Microsoft"-ov paket namenjen primeni računara u kancelarijama.

Očekuje se da će u *MS Office* biti uključena nova verzija *Access*-a koja će nositi oznaku 1.1. U njoj bi trebalo da bude poboljšano povezivanje baza podataka, uključujući i direktni rad sa bazama nastalim u *FoxPro*-u 2.0 i 2.5, *Oracle*-u i *Sybase SQL* serveru. Novi *Access* će podržavati i veće baze nego do sada, maksimalne veličina biće jedan gigabajt, a priča se i o boljoj saradnji sa *Excel*-om i *Word*-om.

Za novu verziju *MS Office*-a je preporučena cena od 949 dolara, odnosno 750 dolara za verziju koja radi pod *Windows*-ima. Recimo još da je od konkurencije već stigao odgovor. „Borland" je počeo sa prodajom paketa u kome se nalaze *Paradox* baza podataka, *Quatro Pro* unakrsne tabele i *WordPerfect* tekst-procesor, i to po ceni od 595 dolara.

Kontakt adresa: Microsoft Corporation, tel. 800-426-9400 i 206-882-8080

HARDVER

Džepni računari

HP 100LX – mali veliki računar

„Hewlett-Packard" je promovisao zamenika svog čuvenog računara koji stane na dlan modela *HP 95LX*. Novi palmtop lični računar *HP 100LX*, težak samo 11 unci koštaće 749 dolara.

U odnosu na prethodni model *HP 100LX* je dobio grafički korisnički interfejs, ispis na ekranu u 80 kolona i veći ROM u koji je smešten softver. I kada smo već kod programske podrške, recimo da će *HP 100LX* raditi pod DOS-om 5.0, biće ugrađene ROM verzije *Lotusa 1-2-3 2.4*, a tu su još i mini tekst-procesor i program za rad sa bazom podataka. Poboljšan je i komunikacioni softver, dodata je mogućnost povezivanja sa „Lotus"-ovim programom *cc:Mail*, a za prenos datoteka mogu se koristiti *Y-modem*, *Z-modem* i *X-modem* protokoli.

HP 100LX ima „Intel"-ov procesor 80C186, koji radi na 7.91 megaherca. Proizvođač tvrdi da su performanse u odnosu na *HP 95LX* poboljšane za oko 50 procenata.

Kontakt adresa: Hewlett Packard Company, tel. 503-750-2038.

Notebook

Za najdublji džep: Moćni IBM notebook

„IBM" je proširio svoju ponudu *notebook* računara sa dva modela. Na vrhu palete sada se nalaze *ThinkPad 720C* i *ThinkPad 720*. Osnovna razlika između ove dve računara je u tipu ekrana – model 720 je sa monohromatskim, a 720C sa kolor ekranom. Recimo još i to da je kolorni displej *ThinkPad*-a izveden u tehnici aktivne matrice i da je sa dijagonalom od 10.4 inča najveći na tržištu.

Zajedničke osobine najsnažnijih „IBM"-ovih notebook računara su procesor IBM 486 LC sa taktom od 50 megaherca, izmenljivi hard-disk kapaciteta 120 MB ili 160 MB (kasnije će biti ponuđena i varijanta od 240 megabajta) i podrška za kartice koje odgovaraju *PCMCIA 2.01* standardu.

Novi *ThinkPad* modeli imaju poboljšani sistem za upravljanje potrošnjom energije. Tako kolor model može neprekidno raditi do 2.4 sata ili kako u „IBM" u kažu do 4.5 sati u tipičnom radnom okruženju. Monohromatski model *720* neprekidno radi 3.5 sati, a u tipičnim radnim uslovima 7.5 sati.

Ako ste zainteresovani za kupovinu ovakvog računara, budite spremni da izdvojite popriličnu svotu novca – treba platiti i dobre performanse, ali i ima firme koja ga proizvodi. *ThinkPad 720C* sa diskom od 160 MB košta 4.695 dolara, dok za model *720* sa hard-diskom iste veličine treba izdvojiti 3.355 dolara.

Kontakt adresa: IBM, tel. 914-251-5959

Laserski štampači

Laserjet 4L – malo je lepo

Teško je zamisliti kompjuteraša koji se ne bi oduševio za novim štampačem HP *LaserJet 4L*. Štampa 4 stranice u minuti odnosno 300 tačaka po inču i neverovatno je mali – visina 6.5 inča, postolje 14x14.5 inča, a teži samo 17.5 funti (8 kg). No, još je manja cena – samo 849 dolara.

Kontrole ovog štampača su sušta jednostavnost. Umesto mnoštva membranskih tastera i LCD ekrana za poruke, tu su samo 4 LED svetlosna indikatora i jedno dugme. Nema prekidača, nema skraćenih poruka – samo jedno jedino veliko dugme.

Ovako pojednostavljen interfejs je dobijen zahvaljujući brojnim novim opcijama koje ima *LaserJet 4L*. Kao prvo, kontroler štampača različito reaguje zavisno od situacije na pritisak dugmeta koje se nalazi sa prednje strane štampača. Na primer, ako u straničnom baferu štampača ima podataka (tada *Ready* i *Data* lampice neprekidno svetle), onda dugme ima funkciju *Page Eject*. Međutim, ako lampica *Data* trepće (što znači da je štampač podešen da prima papir ručno i da čeka da se on ubaci), onda pritisak na dugme ignoriše ovako podešeno stanje i nalaže štampaču da uzme papir iz kaseta.

Drugo, tu je brzi dvosmerni *Bi Tronics* paralelni port, koji ubrzava prenos podataka iz računara do štampača i omogućava da štampač pošalje računaru podatke o svojoj konfiguraciji. Uz ovaj štampač ide uslužni program *HP Explorer* koji zamenjuje meni-sistem na prednjoj ploči, uobičajen kod većine drugih štampača. Možete odrediti *default* fontove i kvalitet štampe pa čak i sačuvati podatke o konfiguraciji za različite aplikacije. Program sadrži i interaktivno *on-line* uputstvo za rukovanje štampačem.

Štampač postiže rezoluciju od samo 300 dpi, ali koristi HP tehnologiju za unapređenje rezolucije (RET – *Resolution Enhancement Technology*), tako da postiže visoki kvalitet štampanog teksta i slika, onakve kakav daje porodica štampača *LaserJet III*. Ima 26 skalabilnih fontova, kao i podršku za *Windows True Type* fontove, uključujući i drajver za *Windows 3.1*.

LaserJet 4L uvodi još jedan novi koncept, takozvani *EconoMode*. U tom režimu, koji podseća na probno štampanje (*draft*), štampa se upola manje tačaka, čime se potrošnja tonera smanjuje za 50%. Rezultat je sasvim jasna slika, ali je boja siva a ne crna kao kod standardnog laserskog otiska.

To nije jedini način na koji *LaserJet 4L* štedi novac. Pošto nema prekidača, štampač se sam uključuje i isključuje. Kada miruje, prelazi u režim manje potrošnje i tada troši oko 5 W – kao obična sijalica. Kada primeti da stižu podaci iz računara, sam se uključi. Nakon završetka štampanja neko vreme sačeka, a zatim posle prelazi u stanje mirovanja.

I pored kompaktnog dizajna i uštede energije, *LaserJet 4L*, sa novom *Canon* mašinom, postiže gotovo istu brzinu kao i veliki i mnogo skuplji štampači *HP LaserJet IIP* i *IIIP* koji koriste *Canon LX* mašinu. Štampa 3.8 stranica teksta u minuti i 1.7 stranica grafika. "Hewlett Packard" napominje da *EconoMode* štedi samo toner a ne i vreme.

U kasetu za papir može stati 100 listova papira, dok se na izlazu prihvata do 50 listova. Otvor za ručno ubacivanje prihvata i koverte. Koristi se jednodelni toner i bubanj s kojim može da se odštampa 3000 stranica.

LaserJet 4L ima 1MB memorije, a može se proširiti samo do 2MB. Međutim, proizvođač pretpostavlja da će proširenje biti potrebno samo u izuzetnim slučajevima, s obzirom da RET tehnologija komprimuje tekst i sliku, tako da slika sa cele stranice zauzima 1MB. *LaserJet 4L* ima podršku za PCL5, a ako tražite više od toga, sačekajte model *LaserJet 4ML* – očekuje se krajem leta (cena 1.279 dolara) i pružaće podršku za PCL i *PostScript*.

Očigledno je da se ovim štampačem "Hewlett-Packard" još jednom dokazuje da malo zaista ume biti lepo. A što je najlepše, uz ovaj mali štampač ide i "mala" cena.

Cena: 849 dolara

Adresa: Hewlett Packard Co., 19310 Pruneridge Ave, Cupertino, CA 95014, 800-752-0900

Matrični štampači

Brži epson LQ-570

Jedan od najpopularnijih „Epsonovih" štampača model LQ-570, od sada se u Engleskoj prodaje u svojoj novoj verziji. Osnovna razlika u odnosu na staru verziju je povećana brzina štampanja, uz istu cenu od 339 funti.

U draft modu *Epson LQ-570+*, kao je ime nove verzije, štampa 266 karaktera u sekundi (ranije 220 pos), dok za kvalitetniji ispis u LQ modu brzina je povećana sa 72 cps na 90 cps. Ne treba zanemariti ni smanjenje nivoa buke štampača koja sada iznosi 47 dB(A).

Ostale mogućnosti *Epsona LQ-570+* „poboljšane mogućnost prihvatanja papira", drugim rečima štampač može da uzima papir iz više izvora bez potrebe da se ručno podešava prelazak sa „fan-fold" na „cut sheet" doturač papira. *LQ 570+* takođe poseduje i još dva nova skalabilna tipa slova.

Monitori

Novi NEC monitori: Jasno kao na dlanu

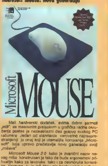
Korporacija „NEC" za svoju reputaciju u PC svetu u velikoj meri može da zahvali svojim monitorima, a po svoj prilici će i dalje ostati među vodećima u ovoj oblasti. Novu generaciju „multisync" monitora čine četiri modela – 3FG, 4FG, 5FG i 6FG, za koje proizvođač kaže da imaju ravne ekrane visokog kontrasta. Svi, osim najmanjeg 3FG, u sebi imaju ugrađenu *AccuColor* kontrolu i „NEC"-ov sistem za „poboljšanu digitalnu kontrolu" (ADC). Monitori imaju veliko iskorišćenje površine ekrana i nisku emisiju zračenja. Od ostalih mogućnosti ističemo frekvenciju osvežavanja ekrana od 70 herca.

Serija FG je optimizovana za prikaz aplikacija zasnovanih na grafičkom korisničkom interfejsu i prikazu boja iz dvadesetčetvorobitne i tridesetdvobitne palete boja. Na monitorima 4FG, 5FG i 6FG, preko ADC sistema i *AccuColor* kontrole, moguće je korigovati odnose između crvene, zelene i plave komponente.

Novi „NEC"-ovi monitori su sa poliranim ravnim ekranima dijagonale od 15 do 21 inča. Rezolucija koju podržavaju je 1024×768 za manje ekrane i 1280×1024 za veće monitore.

Miševi

Microsoft mouse: Nova generacija



Mali hardverski dodatak, svima dobro poznati „miš", sa masovnim prelaskom u grafička radna okruženja postao je nezaobilazni deo gotovo svakog PC računara. Jedan od standarda, verovatno najrasprostranjeniji, je onaj koji je utemeljila kompanija „Microsoft", koja upravo predstavlja novu generaciju svojih „miševa".

Microsoft Mouse 2.0, kako je zvanični naziv novog miša, konstruisan je tako da bude ergonomski prihvatljiv kako za levoruke, tako i za desnoruke korisnike. Naravno, tu je još čitav niz hardverskih i softverskih poboljšanja.

Od poslednjeg redizajna prve verzije „Microsoftovog" miša prošlo je čitavih šest godina. Ovako dugačak, naročito za računarsku industriju, period uzrokovan je odličnom prodajom i čak dvogodišnjim radom na utemeljivanju novih standarda, koje u potpunosti poštuje *Mouse 2.0*. Ponovo je proračunat oblik da bi se optimizovao položaj šake. Masa i položaj kuglice su drugačiji u odnosu na prvu verziju, a izmene koje prate nove ergonomske zahteve doživeo je i drajver za miša, sada u verziji 9.0.

Površina na koju se sada oslanja šaka je duža i veća nego ranije. To doprinosi manjem zamoru korisnika i preciznijem rukovanju. Zadnji deo miša je niži, blago zakošen i zaobljen. Ovo nisu sve promene. Neke od njih, nimalo manje važne, skrivene su ispod plastike. Opto-mehanički sklop koji očitava položaj je precizniji i pouzdaniji, a postavljen je bliže prednjem delu da bi se uspostavila bolja korelacija između položaja kurzora i pomeranja miša.

Što se tiče novog drajvera, glavne izmene su usmerene ka boljoj kontroli i preciznosti postavljanja kurzora na određenu poziciju. Program je sazdan tako da se može koristiti opcija „povećanje" kada se prostor veličine dva inča oko kurzora uvećava i omogućuje korisniku preciznu kontrolu pozicije.

Kontakt adresa: Microsoft GmbH, Edison strasse 1, 8044 Unterschleissheim, tel. 089-3176-38-38, fax. 089-3176-38-30

Milan Bašić i Ranka Jovanović

BORBA ZA VITKIJU LINIJU

Današnji aplikativni softver je nezajažljiv – naprestano traži više RAM-a, više prostora na tvrdom disku, brži mikroprocesor... kao da mu nikad nije dosta. Što je najgora, svi su izgledi da će se ovakva tendencija nastaviti. Glomazan softver i hardverska dogradnja zatvaraju začarani krug. Statistike kažu da korisnik koji kupi novi softver otprilike posle mesec dana traži dogradnju hardvera. A kad dobije novi hardvar, poželi moćniji softver. I tako u krug.

Jednostavne i uobičajene mere za borbu protiv glomaznog softvera su kompresija podataka ili brisanje nepotrebnih programa i modula (vidi antrfile „Kako se boriti protiv glomaznog softvera"). Međutim, pravo rešenje bi bilo kreiranje takvih aplikacija i operativnih sistema koji imaju sposobnost da ponovno koriste isti kod, a ne da programeri moraju uvek iznova da pišu iste funkcije. Platforme kao što su *Windows 3.1*, *Unix*, *System 3* (*Macintosh*) dozvoljavaju korišćenje istog koda, dok objektno-orijentisani operativni sistemi (Apple-ov i IBM-ov *Taligent* i Microsoftov *Cairo*) to obećavaju za početak 1994.

FAKTORI GLOMAZNOSTI

Proizvođač dodaje nove mogućnosti svom softveru da bi održao korak sa konkurencijom, a kupci pomno proučavaju liste mogućnosti pri proceni različitih softverskih paketa. Kompjuterski časopisi samo dolivaju ulje na vatru objavljujući stalno uporedne tabele, gde se lepo vidi šta ko ima a šta nema.

Neki proizvođači su pokušali da ponude „lake" takozvane „lite" verzije, ali korisnici nisu bili baš oduševljeni. *Microsoft Write* i „WordPerfect"-ov *LetterPerfect*, olakšane verzije čuvenih svemogućih tekst-procesora, ne prodaju se baš najbolje, i „Quark"-ov direktor marketinga kaže da korisnici nisu pokazali interesovanje za olakšanu verziju *QuarkXpress* programa za stono izdavaštvo, što je potvrđeno u praksi kada je u jesen 1988. izdat zloglasni *QuarkStyle*.

Proizvođači imaju jak razlog da stalno obogaćuju softver – to je unosno tržište dogradnji. Microsoft je od svake kopije dogradnje za DOS imao prihod od 45-50 dolara, dok je zarada od originalne verzije 20-25 dolara.

Većina velikih proizvođača nudi posebne rutine za instalaciju koje omogućavaju korisniku da instalira samo ono što mu najviše treba. Međutim, problem je u tome što korisnici često ne znaju šta im je zaista nepotrebno, a iz toga proizilazi da svi instaliraju sve po principu „nek se nađe ako zatreba".

PITANJE PROGRAMIRANJA

Sledeći faktor koji značajno doprinosi rastu softvera jeste loše upravljanje razvojem softvera. Dešava se da programeri gube bezbrojne sate da bi „ponovo otkrili točak", a sav suvišni kod završava na vašem tvrdom disku. Aplikacije na čijem razvoju su radili brojni timovi programera imaju mnogo koda koji se ponavlja.

Još jedan kontroverzan aspekt glomaznog softvera je povećana upotreba visokih programskih jezika, C i C++. Dok je asembler u stanju da proizvede veoma sažet kod, u C-u i svakom drugom visokom jeziku će u principu biti više koda. Međutim, dobar programer može porast koda svesti na minimum koristeći sve prednosti visokog jezika.

Između verzija 2.01 i 3.0 programa Lotus 1-2-3 „Lotus Development" je sa asemblera prešao na C – rezultat je bio gotovo utrostručen kod koji je sa 1.4 MB skočio na 4 MB. Za toliki porast naravno nije odgovorna isključivo razlika između asemblera i C-a, mada to jeste glavni razlog. Novi proizvod u tom obliku nije mogao da radi pod DOS-om, tako da je isporuka odložena da bi se izvršila kompresija i optimizacija koda.

Neosporno je da programski kod guta RAM, ali su slike, uzorci dokumenata i pisama, programi za obuku i pomoć u radu još proždrljiviji. Kako programi sve češće sadrže zvučne i video datoteke, potreban prostor na disku dostiže vrtoglave razmere – uobičajeni video-uzorak u trajanju od 15 sekundi traži 2.14 MB čak i u komprimovanom formatu. Što je najsmešnije, većina tih podataka se nikada ne koristi. Ankete pokazuju da korisnici instaliraju hiljade video-isečaka, a upotrebe samo dvadesetak godišnje.

Da bi svoj program učinili lakšim za upotrebu, mnogi proizvođači uključuju *on-line help*. Tako „Microsoft"-ov *Wizards help* zauzima pola megabajta samo za *Access*, multimedijalno izdanje programa *Lotus 1-2-3*, verzija 1.1 za Mek traži oko 3 MB, dok *AutoCad* verzija 11 koja ima dodatak za *Windows* sadrži opcionu verziju uputstva u obliku hiper teksta – zauzima 12MB.

Najčešće pominjani gutač diska je *spelling-checker*. Možete imati poseban *spelling-checker* za tekst-procesor, za program za stono izdavaštvo i za tabelarni proračun, a svaki će potrošiti oko pola megabajta pre nego što dodate rečnika. Kada bi se izdavači *Windows* aplikacija dogovorili da koriste univerzalan rečnik, uštedeli bi nam 2 do 3 megabajta.

Upotreba grafičkog korisničkog interfejsa podstiče nekontrolisano bujanje softvera. Proces razvoja *Windows*-a zahteva resurse koji ne postoje u DOS okruženju. Prozori za dijalog, bit mape i uopšte grafička priroda *Windows*-a guta mnogo RAM-a, mnogo prostora na disku i veliki broj ciklusa mikroprocesora. Uzmimo primer *WordPerfect*-a. Verzija 5.1 za DOS zahteva najmanje 384 RAM i 2 MB na tvrdom disku, dok verzija za *Windows* traži sistem sa barem 4 MB RAM i 4.2 MB prostora na disku. Potreban prostor u stvari iznosi 15 MB, jer se u paketu nalaze i *Adobe Type Manager* i *Grammatik*, program za proveru pravopisa i gramatike. *Motif* verzija *Wordperfect*-a za *Unix* traži 5 MB na tvrdom disku, što je za 66% više od ne-grafičke verzije (vidi sliku 3).

Primećeno je da proizvođači često izbegavaju da daju tačne podatke o sistemskim zahtevima da ne bi oterali mušterije. „Byte" je testirao devet popularnih softverskih paketa u nameri da ispita performanse pri minimalnoj konfiguraciji. Pokazalo se da svi kako-tako rade, ali su performanse drastično smanjene, toliko da je upotrebljivost dovedena u pitanje (vidi antrfile „Preporučeni hardver").

IZGUBLJENA PRODUKTIVNOST

Veliki softver zahteva ogromne ljudske resurse koji se uglavnom ne vide. Zamislite samo koliko se vremena utroši svaki put kada administrator mreže instalira i zatim testira dogradnju za neki program na nekoliko stotina radnih stanica. Instalacija i konfiguracija programa kao što je *Microsoft Word for Windows* može da traje dva sata po računaru. Na to treba dodati još oko osam sati za dodatnu obuku neiskusnih korisnika, ukoliko dogradnja donosi velike promene.

Kompanije obično zanemaruju obuku neophodnu za rad sa složenim aplikacijama. Uvrežena je zabluda da je u *Windows*-u sve očigledno, pa je obuka nepotrebna. Ali korisnici *Windows*-a koji se nikada nisu sreli sa DOS-om doživa pravi šok kad im se prvi put desi da treba da operave, recimo, datoteku c:\mojdir\mojadat.doc.

Ne, obuka korisnika se ipak odvija, ne naročito ali skup način. Korisnici se neformalno uče sami isprobavajući razne opcije i često gubeći mnogo vremena. Jedna studija je pokazala da korisnik prosečno provede 5 sati na učenju, pružanju pomoći kolegama i drugim aktivnostima indirektno vezanim za upotrebu softvera.

blera i C-a, mada to jeste glavni razlog. Novi proizvod u tom obliku nije mogao da radi pod DOS-om, tako da je isporuka odložena da bi se izvršila kompresija i optimizacija koda.

Mnogi korisnici se žale da glomaznost smanjuje upotrebljivost softvera, jer zbog mnoštva raspoloživih opcija nisu usredsređeni na ono što im je stvarno potrebno. Ponuđena im je vreća puna alata, ali što je više alata to je teže odabrati onu pravu.

„Lotus", „Microsoft" i sve veći broj drugih kompanija pokreću vlastite laboratorije za testiranje upotrebljivosti njihovog softvera, u nameri da proizvodi programa učine lakše dostupnim. Većina proizvoda raspolaže potrebnim funkcijama, samo je pitanje koliko se one lako koriste. Međutim, opcije koje treba da unaprede upotrebljivost često i same doprinose glomaznosti koda.

REŠENJA SU U OPERATIVNOM SISTEMU

Glomazan softver je usko vezan za platformu kojoj je namenjen. Ako operativni sistem ne pruži svoje usluge na efikasan način, onda svaka aplikacija mora sadržati dodatni kod.

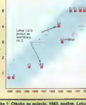
Unapređenje operativnog sistema može učiniti da se softver oslobodi suvišnog koda. Na primer, verzija 2.0 „Microsoft"-ovog prezentacionog softvera *PowerPoint* je prvobitno zahtevala 50 MB prostora na disku ukoliko su se učitavali svi *Bitstream* fontovi. Međutim, pošto *Windows* ima ugrađene *TrueType* skalabilne fontove, *PowerPoint* je spao na svega 14 MB, što znači da je stesao čak 72% svoje „težine".

Sofisticirane novosti koje sadrže *Windows 3.1* i Apple-ov *System 7.1* daju nadu da bi se bujanje aplikativnog softvera moglo staviti pod kontrolu. Naime, ovi operativni sistemi sadrže mehanizme koji omogućavaju da jedna aplikacija traži usluge od drugih i da međusobno razmenjuju informacije. Tako je aplikacija sposobna da obavlja veoma složene zadatke, deleći pojedine poslove sa drugim aplikacijama u sistemu ili u mreži.

Na primer, aplikacija za pripremu stranice za štampu poverava generisanje tabele na osnovu datih numeričkih podataka *spreadsheet* aplikaciji, a izradu grafikona drugoj specijalizovanoj aplikaciji. Kada se suoči sa skeniranom slikom nepoznatog formata, ova aplikacija poziva u pomoć neki grafički program koji konvertuje sliku, skalira je i iseca pre nego što je smesti u dokument.

Ova sposobnost jedne aplikacije da iskoristi umeće drugih trebalo bi da označi kraj ere monolitnog softvera, koji sve sam radi. Umesto da gomilaju funkcije da bi zadovoljili sve eventualne potrebe svakog potencijalnog korisnika, proizvođači će moći da skroje kod prema specifičnim namenama. Samim tim, oni će moći da se skoncentrišu na oblast za koju su specijalizovani. Manji kod znači i manje posla oko otklanjanja bagova i optimizacije za pojedine platforme. Razmena funkcija donosi još jednu veoma dragocenu uštedu – nije potrebno upoznavanje sa radom novih alata.

Windows OLE i za Macintosh *System 7 IAC* (*Interapplication Communication* – međuaplikacijska komunikacija) omogućavaju komunikaciju između aplikacija na principu klijent-server. Ove funkcije nisu kruto postavljene – ista aplikacija koja traži uslugu (dakle, klijent) kasnije može kao server pružiti uslugu nekoj drugoj aplikaciji. Međutim, istovremena klijentska i serverska sposobnost zahteva mnogo posla, tako da će proizvođači verovatno nuditi aplikacije koje mogu služiti samo kao klijent ili samo kao server. *System 7.1* svoju „vikost" duguje integrisanim mrežnim funkcijama, tako da mu nisu potrebni dodatni moduli. Sa druge strane, da bi pomoću OLE tehnike uspostavili mrežnu komunikaciju, morate



Slika 1. Otkako se pojavio, 1983. godine, *Lotus 1-2-3* za DOS je svake dve godine postajao dvaput obimniji – sa flopi-diska od 360K popeo se do 5 MB.



Slika 2 1983 bazična konfiguracija IBM-kompatibilnog PC računara je imala 64 KB RAM i 360 KB prostora na tvrdom disku Danas PC ne bi smeo da ima manje od 4 MB RAM i kapacitet tvrdog diska od 100 MB

imeli *Windows for Workgroups* čak i ako vam ne treba softver za rad u grupi

OLE SVE BOLJI

Uprkos izvesnim nedostacima OLE je najambiciozniija i najšire prihvaćena alatka za komunikaciju između više aplikacija OLE 1.0 radi sporo i ne podržava nevidljivu (*seamless*) integraciju koja bi omogućila da eksterne komponente uspešno konkurišu onim internim ugrađenim Međutim nova izdanje OLE 2.0 čija je beta verzija trenutno na testiranju kod proizvođača softvera a korisnici je mogu očekivati u jesen ime rešenja za oba ova problema

Odbacujući nezgrapni *Windows* DDE protokol i uvodeći novi oblik interprocesnih komunikacija Microsoft želi da obezbedi dovoljno visoke performanse koje će OLE aplikacije učiniti prihvatljivom alternativom za monolitne integrisana programe kakav je na primer Lotus ov *Symphony* Međutim proizvođači smatraju da prelazak na OLE 2.0 nije lak korak Tako direktor proizvodnje programa *CorelDraw* kaže da se svi slažu da OLE pruža lep pogled na svet ali je skok u odnosu na OLE 1.0 suviše nagao

Sa druge strane OLE 2.0 čini život lakšim Pretpostavimo da ste ugradili niz *Lotus 1 2-3* ćelija u *Word for Windows* dokument i želite da izmenite vrednost u jednoj ćeliji Da bi se zadatak obavio OLE 1.0 zahteva dve radikalne promene konteksta prvo *1 2 3* mora preuzeti kontrolu a zatim se one opet mora vratiti *Word*-u Uz OLE 2.0 funkcije *Word*-a i *1 2 3* se privremeno izmešaju dok su ugrađene ćelije u fokusu a posle toga *Word* opet poprima svoj prirodni oblik

No čak i ako OLE 2.0 opravda sva očekivanja ova tehnika će ponuditi tek početnu osnovu za rešavanje problema glomaznog softvera Pravi izazov će biti izgradnje komponenti koje će moći da se priključuju po potrebi Današnji integrisani paketi nude ubedljive prednosti Kod programa *Interleaf Publisher* na primer svi objekti dele zajedničku prototokole tako da se promena si te ošćuje u celom dokumentu gde se na adekvatan način menjaju tekst grafika Ova originalna objektna arhitektura je specijalnost *Interleaf*-a i njegova ključna prednost nad ostalim programima slične namene

Iako bi OLE koncept „pregovaranja o svojini" mogao omogućiti da *Interleaf* ustupa svoje usluge druga aplikacija bi morala strahovito dobro poznavati *Interleaf* komponentu da bi je mogla koristiti podjednako efikasno kao neki *Interleaf* klijent Što je potrebno poznavanje veće specifičnije to je manja verovatnoća da će neka aplikacije biti u stanju da slobodno upada u daljove programa drugog proizvođača.

Vrlo je verovatno da ćemo uskoro dobiti porodice proizvoda koji koriste zajedničke protokole Osim toga javiće se proizvođač koji će isporučivati isključivo komponente Dobri primeri već postoji *Pinnacle Graphics Server* (proizvođač Pinnacle Publishing DLL odnosno biblioteka dinamičkih veza koju koristi nekoliko *Windows* baza podataka za grafički prikaz ili *Data Junction* (Tools & Techniques) univerzalni program za konverziju iz jednog formata u drugi

KADA VEĆE ZNAČI BOLJE

Obiman softver nije uvek loša stvar Neki faktori koji program čine glomaznim donose korisniku dugoročne prednosti robustan kod kod lak za održavanje kao i sposobnost aplikacije da se izvršava na više platformi

blera i C-a mada to jeste glavni razlog Novi proizvod u tom obliku nije mogao da radi pod DOS om tako da je isporuka odložena da bi se izvršila kompresija i optimizacija koda

Tko na primer proizvođač može odlučiti da u projektovanju aplikacije primeni objektno programiranje Jedna od prednosti objektnog programiranja (OOP) jeste mogućnost ponovnog korišćenja istog koda To u krajnjoj liniji smanjuje troškove razvoja i održavanja OOP uglavnom koristi visoke programska jezika a objektni kod koji ovi jezici (npr C++) generišu nije tako kompaktan kao onaj pisan u asembleru Međutim baš zato što je pisanje koda postalo spor i skup proces mogućnost koju nudi OOP da se brzo isprave bagovi i dopuni aplikacija daleko je važnije od činjenice da je takav kod uzgred i neizbežno glomazan

Aplikacije namenjena radu na više platformi kao što je *Aldus PageMaker* ili *Adobe Photoshop* obimnija je naravno od aplikacije pisane za jednu platformu Kada se pojavio 1989 godine *Photoshop* program za obradu slika je bio *Pascal* program za *Mac* računare Kada je došlo vreme da se predaci i na *Windows* aplikacija je potpuno redizajnirana i napisana na jeziku C++ Obilato je korišćen OOP kako za apstrakciju podataka tako i za primenu ponovne upotrebe koda Većina koda je napisana u obliku visoko apstrahovanog elementarnog *(core)* koda

Elementarni kod se zove tako zato što implementira elementarne operacije programa u slučaju *Photoshop*-a to je konverzija modela za boje upotreba filtara za slike manipulisanje kanalima editovanja piksela, i druge Elementarni kod se ne bavi korisničkom interfejsom čije su operacije specifične za svaku platformu Upotreba elementarnog koda je omogućila autorima *Photoshop*-a da većinu elementarnih funkcija napišu u prenosivom C++ kodu Time je za prenošenje *Photoshop*a na drugu platformu trebalo još samo izvršiti kompajliranje elementarnog koda na ciljnoj platformi i obaviti ga slojem visoko optimizovanog koda za korisnički interfejs (za *Mac* ili za *Windows*)

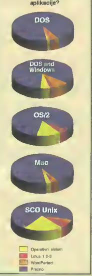
Po rečima Stiva Gutmana, direktora marketinga za *Macintosh* platformu u kompaniji „Adobe Systems", ova tehnika nudi izvanrednu uštedu Kao prvo kod se piše samo jednom Kao drugo bag ispravljen u elementarnom kodu se odnosi na sve platforme jer sve koriste isti kod Kao u gore navedenom primeru za OOP elementarni kod je pisan visokim jezikom koji se ne oslanja na bilo kakav specifični hardver Konačni ishod jeste veća količina objektnog koda ali koda koji omogućava implementaciju i podršku na više platformi

Drugi faktor koji doprinosi obimnosti aplikacija za više platformi jeste činjenica da je neizbežno dodavanje koda kako bi se prevazišla izvesna ograničenja platforme i obezbedila opcija koje joj nedostaju Tako su zbog jednog nedostatka u *Windows* mehanizmu za upravljanje memorijom programeri *Photoshop*-a za *Windows* platformu morali da napišu novi *handler* memorije

Što se budućnosti tiče proizvođači softvera se pripremaju za prelazak na RISC mikrokompjutere koji upravo izlaze na tržište Za sada *Windows NT* ima podršku za Mips-ov R4000 DEC ov *Alpha* i Intel ov *Pentium* *Windows NT* se očekuje tokom ove godine a proizvođači žele da imaju spremne aplikacija u času kada se pojavi novi operativni sistem

Planiranje prelaska na novi operativni sistem utiče na dizajn programa Uobičajeno je da se program prvo u potpunosti napiše nekim visokim jezikom Posle toga tim programera u asembleru ponovo piše neke segmente bržim asemblerskim kodom Optimizacijom ključnih segmenata koda postiže se dramatičan skok performansi bez svođenja celog programa na jezik asemblera Na žalost ista ova optimizacija vezuje aplikaciju za određeni hardver Da bi privremeno izbegao ovaj problem proizvođač se može odlučiti da ne odavi doradu koda pisanog visokim jezikom i tako omogući bržu transformaciju koda za prelazak recimo na RISC mašinu Ovakav postupak ima za rezultat veću i sporiju aplikaciju ali je proizvođač spreman da brzo prilagodi kod novoj tehnologiji kada za to dođe vreme

Dakle porast koda ne mora da bude odraz gomilanje nepotrebnih opcija već rezultat nastojanja da se korisniku pruži upravo ono što želi stabilan kod brzo ispravljanje bagova i rad na više platformi



Slika 3 Prostor na disku koji zauzimaju aplikacije i operativni sistem na hard-disku od 200 MB varira u zavisnosti od platforme Podaci za operativni sistem se odnose na punu instalaciju

ALDUS ČINI PRVI KORAK

Pitanje je međutim zašto bi proizvođači prestajali na primenu OLE funkcije? Na tržištu na kojem ima nekoliko

PREPORUČENI HARDVERSKI RESURSI

Sudeći po onome što piše na kutiji, *Microsoft Access* se izvršava na svakoj mašini klase 386 koja ima najmanje 2 MB memorije i 8 MB slobodnog prostora na tvrdom disku. I zaista, to je izvodljivo – pod uslovom da se odreknete *SmartDrive*-a, da naterate *Windows 3.1* da radi u *enhanced* režimu na sistemu od 2MB, i da imate još nekoliko dodatnih megabajta na disku za *swap* datoteku. Ali, kad konačno podesite konfiguraciju i uspete da pokrenete aplikaciju koja je poslednja reč tehnologije na računaru koji je proizveden još pre dva godine – dokle možete stići?

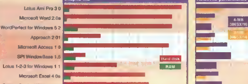
Ukoliko ste bezgranično tolerantni, moći ćete za… da obavite bilo koji zadatak. Ali ako vam strpljivost nije jača strana, onda ćete imati naprika. BYTE je pokrenuo seriju aplikacija na mašini 486DX/33 sa [illegible] RAM, i na istoj 386SX mašini ali sa 4 MB RAM. Namera je bila da se uporede performanse tipične moderne konfiguracije i one koja jedva zadovoljava minimalne hardverske zahteve. Sve testirane aplikacije su se izvršavale na minimalnoj konfiguraciji, ali obično deset puta sporije nego na mašini 486DX/33.

Naravno, manje procesorske snage znači sporije pristupanje memoriji, niska memorija povlači za sobom česte prelaske na spori tvrdi disk. O razlici rečito govori iznenađujući odnos performansi na tipičnoj, odnosno minimalnoj konfiguraciji.

Pokretanje velikih aplikacija na minimalno konfigurisanom sistemu je krajnje razočaravajuće. Potrebno je, na primer, više od jednog minuta da bi se samo startovao *Microsoft Access*. U programu *WindowsBase* (Software Products International), ako pogrešno odaberete aktivnost, moraćete čekati petnaestak sekundi da se vrati pointer miša. Skrolovanje stranice u *Wordperfect*-u za *Windows* može potrajati i desetak sekundi. Kod interaktivnih aplikacija, vreme odgovora (*response*) je kritično. Kada vreme odgovora *low-end* sistema padne na red veličine manji od tipičnog sistema, onda interaktivni softver postaje neupotrebljiv.

Dakle, iako teoretski izvodljivo, pokretanje *Windows* aplikacija na minimalnoj konfiguraciji daleko je od preporučljivog. Ili dogradite mašinu ili zaboravite na najnoviji softver, jer sa kombinacijom najnovijeg softvera i zastarelog hardvera nećete daleko dospeti.

WINDOWS APLIKACIJE PRI MINIMALNOJ KONFIGURACIJI



Podaci o prostoru na disku se odnose na punu instalaciju. Zauzeta memorija uključuje i otvoreni test-dokument. Podaci o performansama su dati kao procenat performansi na naprednijem sistemu (486DX/33 sa 16 MB RAM). Četiri aplikacije – *SPI WindowsBase*, *WordPerfect for Windows*, *Lotus 1-2-3 for Windows* i *Quattro Pro for Windows* – zahtevaju minimalnu konfiguraciju od 3 MB ili 4 MB RAM, zato za njih dajemo samo performanse postignute na sistemu od 4 MB.

programa za tabelarne proračune, paketa za crtanje procesora, programa za upravljanje bazama podataka, nastoji da se njegov proizvod razlikuje od drugih. Svoaplikacije na osnovne funkcije i stavljanje tih funkcija raspolaganju drugim programima, uopšte nije tržišno le… kancepcija.

pak, prve korake na tom polju je preduzeo „Aldus". U je predstavljena verzija programa *Persuasion* čija je sa za pripremu grafikona svedena na jednonamensku stalnu aplikaciju, takozvani „applet" ili „programčić" e kao OLE server i koristi OLE 1.0, pripremajući gra- e za aplikacije koje zatraže njegove usluge. Kada jed kupite *Persuasion*, njegove usluge će biti dostupne OLE klijentima.

Ovakva implementacija „Aldus" ovog programa za grafizgleda kao razuman kompromis – na tržištu se nudi rske aplikacije, ali bez dragulja iz krune koji su saču u originalnom velikom paketu. Idealno bi bilo da svi ređači prestanu da pišu praktične klijentske i serverske acije koje bi se mogle razmenjivati. Tada bi korisnik o slobodno da bira program pomoću kojeg će najbolje akše obaviti posao.

Međutim, vrlo je verovatno da će proizvođači softvera ti privatne OLE poruke za intenzivnu komunikaciju meim vlastitim aplikacijama, dok će komunikacija sa sof- m drugih firmi biti slaba ili nikakva. Tada biste morali odlučite da sva aplikacije kupite od istog proizvođača biste u potpunosti iskoristili sve što mogu da pruže.

Ovaj poslednji scenario ima najviše izgleda da se ostva- tehnička i marketinška stvarnost u kompjuterskoj inji je takva da upotreba modularnog softvera nije nimalo čna. Dakle, proizvođači će verovatno podržati gole kotacione standarde, ali će zato izdavati dogradnja

MERE PROTIV GOJAZNOSTI

Devet koraka koje predlažemo da preduzmete nisu lek već samo terapija koja će vam pomoći da sa glomaznim softverom lakše živite.

Kupite što više brzine, RAM-a i prostora za skladištenja prilikom nabavke sistema, jer će vas kasnije dogradnje skuplje koštati.

Kupite integrisani softver koji u jednom paketu nudi sve uobičajene funkcije – tekst-procesor, upravljanje bazom podataka i program za tabelarni proračun (osim, naravno, ukoliko vam je potreban program specijalizovan za jednu namenu).

Kupite CD ROM drajv, na kojem možete čuvati fontova ili biblioteka simbola.

Centralizujte, po mogućnosti, što veći broj aplikacija – smestite ih na veliki mrežni server.

Arhivirajte datoteke – one koje se često koriste uz pomoć transparentnog (*on-the-fly*) kompresora, a podatke koji se koriste retko uz pomoć arhivera.

Obrišite programe za obuku (*tutorial*) kada savladate program.

Obrišite nepotrebne programe koje ste dobili u paketu, kao i njihove help-datoteke.

Identifikujte svoje stvarne potrebe i proverite da li im ponuđena dogradnja odgovara pre no što se odlučite za naprednju verziju.

Obratite pažnju na preporuke o kapacitetu diska i memorije koje se daju uz softver.

OBJEKTNA BUDUĆNOST

Ipak, najviše nade uliva dolazak objektnih operativnih sistema. Novi operativni sistemi, IBM-ov i Apple-ov *Taligent* [illegible], nudeći korisniku niz objekata koji će moći posebno da se kupe, dograde i povežu po potrebi.

Kako će izgledati ovi objektni operativni sistemi, možemo videti već sada, na primeru *Oberona*, novog operativnog sistema i programskog jezika koji je razvio autor *Pascala*, profesor Niklaus Wirth. On smatra da je današnji softver postao preglomazan, neefikasan, pa čak i nepouzdan. Već su spremne verzije *Oberona* koje se izvršavaju na raznim platformama – PC, Mac, Sun Sparcstations i IBM RISC System/6000S.

Wirthovo rešenje je proširivost. Uz pomoć programskog jezika *Oberon* korisnik može unapred prilagoditi svojim potrebama jednostavne alatke, kao na primer editor teksta. Ovako prilagođena verzija će zauzimati manje prostora, jer koriste isti bazični kod za editor teksta sadržan u operativnom sistemu.

Druga važna osobina *Oberona* je način na koji se vrši pregled podataka. *Oberon* ima „prikazivače" (*viewers*) u kojima možete pregledati dokument. Dokument se sastoji od teksta i grafika i sadrži komande kojima se menja sadržaj dokumenta. Kada dođe do promene, dokument šalje poruku svim prikazivačima, tako da mogu ažurirati prikaz. To što su dokument i prikazivač razdvojeni, omogućava korisniku da kreira nove tipove dokumenata i prikazivača, a da se pri tom ne dupliše postojeći kod operativnog sistema.

Iako je *Oberon* kompletan grafički operativni sistem, sa vlastitim kompajlerom i tekst-editorom, njegova verzija za PC zahteva samo 1.5 RAM, uključujući i izvorni kod. Poređenja radi, *Windows 3.1* zauzima 9 MB.

U NAJTEŽOJ KATEGORIJI

Konačno smo došli u priliku da vidimo na delu najnoviju seriju procesora iz Intelove i486 familije (DX2/33-66 i 486/50 MHz), najnoviju generaciju matičnih ploča sa lokalnom sabirnicom u VESA standardu i najnoviju generaciju grafičkih kartica i kontrolera tvrdog diska projektivanu upravo za VESA sabirnicu. Prvi utisak je da nova PC arhitektura ispunjava svoja spektakularna obećanja.

Sredinom maja, u redakciji se našlo nekoliko 486 mašina najnovije generacije. Sve u svemu, po tehničkim specifikacijama i poreklu robe dosta mešani harver: jedna mašina (Photo, vlasništvo foto-studija Tetac) 486DX2/66 sa potpuno klasičnom ISA arhitekturom, jedna (ADA) 486DX/66 sa VESA lokal bas arhitekturom, VESA video karticom i ISA kontrolerom tvrdog diska, jedna klasična (INFO-D) 486/55 MHz mašina sa ISA arhitekturom i tri VESA 486/50MHz mašine sa lokalnim basom (ComTrad, Micro 486, MP Biro) sa VESA kontrolerima tvrdog diska i VESA video karticama. Ovoj reprezentativnoj zbirci odmah smo dodali i jednu redakcijsku (ComTrad) 486/33 MHz mašinu da bismo u igru uveli i reper koji je nama, a verujemo i izvesnom broju čitalaca, dobro poznat.

IZJAVA O NAMERAMA

Bila je to prva prilika da vidimo na delu najnoviju generaciju procesora iz Intelove i486 familije (DX2/33-66 i 486/50 MHz), najnoviju generaciju matičnih ploča sa lokalnom sabirnicom u VESA standardu i najnoviju generaciju grafičkih kartica i kontrolera tvrdog diska projektivanu upravo za VESA arhitekturu personalnih računara. Umesto testiranja kompletnih konfiguracija (u našim uslovima pojam „konfiguracije" ima prilično rastegljivo značenje – monitori, tastature, miševi i tvrdi diskovi su po principu „daj šta ima" krajnje promenljiva stvar) svu pažnju smo usredsredili na to koliko brži procesori i VESA arhitektura poboljšavaju ključne podsisteme PC mašina i koliko doprinose njihovim performansama u celini. Procesor DX2 koji „iznutra" radi na 66 a „spolja" na 33 MHz posebno je zanimljiv zbog svog dvojnog takta, a lokalna sabirnica zbog svojih ambicija da, obezbeđujući direktan priključak na magistrale procesora, spektakularno poboljša performanse dva najkritičnija podsistema u personalnom računaru – za skladištenje velikih količina podataka (tvrdi disk) i njihovo prikazivanje (video stepen). O VESA standardu i lokalnoj sabirnici smo do sada pisali u više navrata, pa vas za detalje upućujemo na ranije brojeve „Računara".

Za ispitivanje mašina smo koristili tri vrste testova. Njihovo ponašanje pod DOS-om smo posmatrali kroz čuvene *PC Benchmark* testove (verzija 7.0), dok smo se pod *Windows*-ima opredelili za *PC Winbench* i *Wintach*. *PC Benchmark* je u potpunosti mašinski test i obuhvata četiri elementarna resursa (procesor, memoriju, video i tvrdi disk), ali su ispitivanja izabrana a rezultati izvedeni tako da oponašaju najznačajnije PC aplikacije – način na koji one pristupaju i koriste resurse mašina. Stoga se smatra da postoji prilično tesna veza između relativnih onih koje su dobijene na testu i stvarnih performansi mašina, dakle njihovog ponašanja dok izvršavaju programe. *PC Winbench* predstavlja kombinaciju

Jovan Regasek

mašinskih i aplikativnih testova (mi smo koristili samo onaj deo koji se odnosi na grafičke kartice), dok *Wintach* (test je razvila firma Texas Instruments za testiranje svojih grafičkih procesora) obavlja aplikativne testove video podsistema za četiri najznačajnije kategorije PC softvera – obradu teksta, CAD crtanje, slikanje (*paint*) i rad sa tabelama (*spreadsheet*). Pored zbirnih opštih ocena mašina, dajemo i rezultate u pojedinim kategorijama. Iz njih se dobro vide jake i slabe strane testiranih platformi i mogu se izvući dosta pouzdani zaključci o tome za koju vrstu računarskih poslova je koja među njima primerenija.

Prilikom testiranja su, razume se, poštovani oni uslovi koje zahtevaju pomenuti testovi, ali se nismo upuštali u proveru podešenosti mašina. Pošli smo od pretpostavke da su ih njihovi vlasnici već podesili najbolje što se može, iako rezultati pokazuju da ipak nisu. Svaka mašina je došla u kompletnoj konfiguraciji, ali smo mi sva testiranja izvršili sa jednim monitorom – Eizo FlexScan 540i (MICRO) sa ravnim ekranom dijagonale 17 inča i veličinom tačke od 0.26 mm – čije su nam izuzetne performanse omogućile da vidimo novu generaciju grafičkih kartica na delu u punom sjaju, dakle sa kompletnom paletom boja, kristalno jasnom prirodnom slikom i izuzetno oštrim i kontrastnim ispisom teksta, čak i u rezolucijama 1280x1024. Testove Ada 486/DX2 mašine smo otpočeli takođe sa jednim monitorom od 17 inča, ali je on prebrzo prošao kroz redakciju da bismo uspeli da zapazimo njegove osobine. Photo 486 je testiran sa sopstvenim Mitsubishi monitorom od 17 inča, takođe veoma dobrih performansi. Zbog nedostatka odgovarajućih uslova, sami monitori nisu testirani.

MATIČNE PLOČE

U centru naše pažnje su se tako našle dve komponente računara – procesor sa svojim prirodnim okruženjem, odnosno kompletnom matičnom pločom, i grafička kartica kao najproblematičnija komponenta sistema u CAD aplikacijama GUI tipa i multimediji.

Nikada, ako ni zbog čega drugog ono iz marketinških razloga, nismo razumeli tajvansku *no name* filozofiju kada je u pitanju tako važna komponenta računara kao što je matična ploča. Bilo kako bilo, na tri matične ploče (iako da ih je uradila ista ruka, ali ih odaje *isto* uputstvo, mada se radi o dve *generacije* – jednoj ISA iz 1991. i drugoj VESA sa lokalnom sabirnicom iz februara 1993.) nismo uspeli da pronađemo nikakav trag o njihovom proizvođaču. Prvi tip je ugrađen u INFO-D, a drugi, u potpuno redizajniranoj verziji, u ComTrad i MP Biro mašine. Rado bismo vam opisali ovaj tip ploča jer su se pokazale sasvim dobro, ali ne znamo na osnovu čega biste ih prepoznali osim možda po tome da koriste SiS kontrolnu logiku.

Najviše pažnje je, po prirodi stvari, izazvala matična ploča sa kriptografskim imenom ABC (A4883) jer je njen konstruktor najviše mislio na budućnost. U njoj se nalazio procesor i486DX2, ali u tzv. ZIF podnožju (za lako ubadanje) koje može da primi i druge tipove procesora, između ostalog i čuveni Pentium – onda kada se bude pojavio. Ploča koristi Forex (FRX46C421 i FRX46C421) kontrolnu logiku i najnaprednije tehnike za upravljanje radnom i keš memorijom (brzi stranični upis i potpuno transparentno osvežavanje memorije), ali na testovima nije ostavila najbolji utisak. Ne znamo kako bi se na ovoj ploči osećao Pentium, i486DX2 je imao izvesnih teškoća u saradnji sa ostalim delovima sistema: sekundarnim kešom, memorijom i video karticom. Ploča ima dve 32-bitne VESA sabirnice.

Matična ploča HOT-409 486/DX iz februara 1993. ugrađena u računar Micro 486 ima dve 32-bitne VESA lokalne sabirnice, kontrolnu logiku obezbeđuje baterija OPTI čipova i ničim se ne izdvaja iz standarda. Konačno, u Photo 486 je ugrađena NE3486 Multi matična ploča sa Forex kontrolnom logikom i ISA sabirnicama.

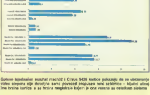
Matične ploče su ili četiri *baby* dimenzija ili, „iz razloga pouzdanosti", kako navode njihovi proizvođači, 2/3 *baby*, ali su u oba slučaja dovoljno male da stanu čak i u notebook računar. Stoga se osećaju veoma usamljeno u ogromnim *tower* kućištima čiji gabariti zvrje uglavnom prazni i u kojima ima dovoljno prostora za obilje specijalne opreme. Na velikim brzinama procesori oslobađaju ogromnu količinu toplote, pa su svi dobro zaštićeni masivnim hladnjacima, a jedan čak i sopstvenim ventilatorom snage 1W. Ovo rešenje nam je izgledalo sumnjivo (šta ako ventilator zariba?) sve dok, otvarajući mašine, nismo naišli na ploču sa procesorom bez hladnjaka – dobar znak da se čak i u slučaju da zakaže hlađenje, skupocenom procesoru neće desiti ništa neprijatno. Dinamičke memorije brzine 60 ns u radnoj memoriji i statičke brzine 20 ns u sekundarnom procesorskom kešu, predstavljaju standardnu i sasvim uobičajenu sliku za ovu klasu platformi.

GRAFIČKE KARTICE

Proizvođači grafičkih kartica ne pate od anonimnosti kao proizvođači matičnih ploča, uglavnom zato što dolaze sa drugog kraja sveta, na kome se najviše drži do poslovnog imena. Sve u mašinama što pored njih, leži u redakciji sa



Fotografije: Studio Talić & Nenad Petrović



Gotovo izjednačen rezultat mach32 i Cirrus 5426 kartice pokazuju da se ubrzavanje video stepena nije dovoljno samo povećati propusnu moć sabirnice – ključni uticaj ima brzina kartice, a ne brzina magistrale kojom je ona vezana sa ostatkom sistema.

našlo nekoliko čuvenih modela. „Računari" su dosta pisali o novoj generaciji grafičkih kartica pa vam za tehničke detalje preporučujemo tekst „Pravi izbor u eri promene" iz broja 91

ATI mach32 Graphics Ultra+ kartica je jedan od najslavnijih grafičkih akceleratora na tržištu. Projektovana je posebno za primenu sa CAD i *paint* programima, dakle za rad sa softverom za slikanje i obradu fotografija u boji. I u pogodnostima zaista ide barem jedan korak ispred konkurentskih proizvoda na tržištu. Kartica nudi 16,7 miliona boja ne samo u rezoluciji 640x480 nego i u 800x600, i dok druge kartica nude 1024x768 i 1280x1024 u 256 i 16 boja, mach32 obezbeđuje 65000 i 256 boja respektivno. Kartica ima linearnu video memoriju kapaciteta 2 MB, koja se koristi za sliku, grafički keš i virtuelni ekran u kome se može čuvati slika znatno većih dimenzija od onih koje je moguće prikazati na ekranu, a pomeranje prozora po ovoj slici je izuzetno brzo i glatko. Ovu osobinu smatramo dragocenom u radu sa velikim crtežima i fotografijama

ATI mach32 (Photo 486DX2) je sagrađen oko sopstvene grafičke mašine, ima 32-bitnu internu magistralu za podatke, 24-bitni RAMDAC – digitalno-analogni konverter u kome se mešaju boje i čitav niz elementarnih grafičkih operacija ugrađenih u hardver. Proizvodi se za sve tipove sabirnice, a mi smo radili sa ISA modelom. Kartica dolazi sa drajverima za sve iole značajnije aplikacije, uključujući, dakako, i *AutoCad* displej list drajvere (*DL/Xpres*), kao i sa sopstvenim fontovima. Kartice prepoznaje *TrueType* fontove, koje generiše u letu, ali joj prikaz fontova nije baš jača strana.

Kartica HG426AV je sagrađena oko Cirrus Logic 5426 BitBLT grafičke mašine, koja radi na 50 MHz i predviđena je za povezivanje na VESA lokalnu sabirnicu. Ima elementarne grafičke funkcije u hardveru. Integrisani RAMDAC sa 32-bitnom magistralom i 1 do 2 MB linearne dinamičke memorije. Podržava standardne rezolucije od 640x480 u 16,7 miliona boja do 1280x1024 (sa preplitanjem) u 16 boja. Ova grafička kartica je bila ugrađena u tri mašine – ADA 486DX2, Comtrad 486/50 i Micro 486/50

Kartice Diamond Stealth 24 i Stealth Pro (Micro) pripadaju istoj porodici i veoma su srodne građe. Obe su sagrađene oko S3 čipa, prva oko starijeg i sporijeg 86C801 (80 MHz), a druga oko modernijeg i bržeg (135 MHz); prva je predviđena za ISA, a druga za VESA lokalni bas. Podržavaju palete do 16,7 miliona boja i rezolucije do 1280x1024 u 16 boja. Prva ima 8-bitni, a druga 64-bitni RAMDAC

Sve VESA kartice, i HG426 AV i Stealth, su opremljene čitavom baterijom drajvera za sve značajnije operativne sisteme i aplikacije. *AutoCad* programi su podržani odgovarajućim *dipley list* drajverima. Instalacija drajvera za *Windows* je laka i potpuno bezbolna – ako se odabere *Multi-Resolution* drajver, grafički režim se može menjati pomoću posebne alatke bez mukotrpnog odlaska u *setup*.

UBRZAVANJE POČINJE...

Najmoćnije PC mašine na tržištu trenutno se grade oko dva modela Intelovih procesora. Prvi je „frizirani" 486 na 33 MHz koji koristi tzv. „overdrajv" tehnologiju i koji radi na dva takta – interne operacije obavlja brzinom od 66 MHz, a spoljašnje brzinom od 33 MHz. S obzirom da su „prosečne brzine" ova dva tipa procesora matematički jednake, kako se ponašaju na procesorskim testovima i koliko stvarno doprinose performansama čitavog sistema?

Između radnog takta i performansi procesora postoji, kao što bi se i očekivalo, linearna zavisnost. Procesori 486 čiji je časovnik navijen na 50 MHz taman su onoliko brži od mašine na 33 MHz koliko je 50 veće od 33. Iako testovi koje smo koristili ne mere golu brzinu procesora, nego su podešeni tako da simuliraju ponašanje popularnih aplikacija i, pored samog procesora, obuhvataju interfejs procesor/memorija, emulaciju matematičkog koprocesora i sam matematički procesor, sve četiri „pedesetice" daju gotovo u dlaku iste rezultate. To znači da je na svim ovim pločama mikroprocesor dobro uklopljen u sistem i da odlično sarađuje sa ključnim delovima mašine. Kod DX2 klase, na dve različite matične ploče pojavile su se prilično uočljive razlike. Procesor na matičnoj ploči ABC A4683 (Ada) daje nešto slabiji rezultat od DX2 procesora na ploči NE3466 (Photo), uglavnom zbog poteškoća u komunikaciji sa sekundarnim kešom, ali su obe primetno brže od čistih „pedesetica"

Uticaj takta procesora na performanse mašine je već delikatnije pitanje. DX2 procesori su, na procesorskim testovima, definitivno brži od procesora na 50 MHz, ali ipak daju znatno skromniji doprinos ukupnim performansama računara. Na te performanse bitan uticaj ima i komunikacija sa radnom memorijom, video podsistemom i diskom, a sa ovim delovima sistema DX2 i dalje komunicira brzinom od 33 MHz. Intel za svoje *overdrive* procesore navodi dobitak u aplikativnim performansama od 70% u odnosu na čistu 33



Ada Computers i486/DX2 33-66 MHz: Midi *tower* kućište sa prostorom za dve jedinice 3,5 i tri jedinice 5,25 inča i napajanjem snage 200 W. Matična ploča tipa A4683 sa dve VESA lokal bas sabirnice, ZIF podnožjem za nekoliko tipova procesora, uključujući i Pentium. 256 K sekundarnog keša i 8 MB RAM-a proširljivo do 32 MB, i Eorex kontrolnom logikom. VESA lokal bas video kartice tipa HG426AV sa 2 MB RAM-a. Tvrdi disk ALPS kapaciteta 200 MB sa ISA kontrolerom. Ugrađen strimer Jumbo 250 MB. Kolor monitor Casper od 14 inča. **Namena:** univerzalna primena. **Ocena:** Matična ploča zanimljivog dizajna ali sa performansama nešto ispod očekivanja, tvrdi disk izrazito spor. **Kontakt adresa:** *11000 Beograd, Tadeuša Košćuška 72, tel 011/186-355, fax 186-267*

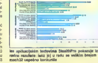
MHz, ali mi sa onim mašinama koje smo imali na testu nismo uspeli da postignemo takav rezultat. Jedna od njih u tehnologiji iz 1993. je nešto sporija od 486/33 MHz redakcijske mašine u tehnologiji iz 1991, dok je druga negde na pola puta između 33MHz i 50MHz mašine. DX2 mašine dolaze na svoje kod aplikacija koje se više oslanjaju na numerički aparat procesora nego na njegovu sposobnost da manipuliše velikim količinama podataka. Na *LinPack* testovima (videti poseban članak u ovom broju) koji su namenjeni za ispitivanje upravo numeričkih performansi računara DX2 mašine zauzimaju listom sva prva mesta i za određene kategorije korisnika sigurno predstavljaju privlačnije rešenje

U SVETU DOS-a

Iz memorijskih testova se verovatno mogu izvući najmerodavniji zaključci o kvalitetu dizajna matičnih ploča. Oni obuhvataju komunikaciju na najosnovnijem nivou sa konvencionalnom, proširenom, grafičkom, tekstualnom i BIOS memorijom, a na rezultate, pored procesora, najdirektnije utiču specifična rešenja u arhitekturi matičnih ploča, kao što su



Mašine imaju ujednačenu prosečnu ocenu, ali analiza ključnih podsistema otkriva suptilnu razliku



Na aplikacijskim testovima StealthPro pokazuje izvanredne rezultate, iako joj u radu sa velikim brojem boja mach32 uspešno konkuriše



MP Biro i486/50 MHz: *Tower* kućište sa prostorom za šest jedinica 5,25 ili 3,5 inča i napajanjem snage 230 W. Matična ploča *no name* tipa sa dve VESA 32-bitne lokal bas sabirnice, 256 K sekundarnog keša i 20 MB RAM-a, proširljivo do 32 MB, sa SiS kontrolnom logikom. VESA lokal bas video kartica tipa HG426AV sa 2 MB RAM-a, VESA lokal bas IDE disk kontroler sa 4 MB keša i mogućnošću priključenja do 2 IDE diska. Tvrdi disk WD Caviar 2250 kapaciteta 250 MB. Kolor monitor Adonic od 14 inča. **Namena:** server za veliku mrežu. **Ocena:** podjednako dobri podsistemi se uklapaju u skladnu celinu, a tvrdi disk, zahvaljujući keš kontroleru, izrazito odskače. **Kontakt adresa:** *11000 Beograd, Beogradska 41, tel 011/341-832, 495-361*

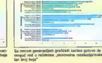
brzi stranični pristup podacima u memoriji i potpuno transparentno osvežavanje memorije kao i brzina odziva memorijskih čipova. Na testu su ploče pokazale dramatične razlike, pa čak i zbunjujuće rezultate, ali njihov uticaj na opšte performanse mašine nije odlučujući i razlike se prilično potiru u konačnom zbiru

Zanimljivo je da su u ovoj kategoriji testova najbolji i najgori rezultat postigli DX2 računari, računajući i kontrolnu 33MHz mašinu. Svoje poene za prvo mesto PHOTO 486 DX2 je sakupio u komunikaciji sa proširenom, konvencionalnom i BIOS memorijom, dok je, zbog ISA video kartice, potpuno podbacio sa grafičkom i tekstualnom memorijom. Žao nam je što nismo imali barem još jednu DX2 mašinu na testu. Na osnovu dobijenih rezultata ne možemo da zaključimo koja je klasa mašina zaista brža u komunikaciji sa memorijom – DX2/66 ili DX/50, odnosno da li je postignuti rezultat izuzetak ili pravilo i u kojoj meri na njega utiče podešenost mašine, posebno broj taktova za usporavanje pristupa (*wait state*)

Nijednom podsistemu PC računara nisu bila potrebna toliko hitna unapređenja kao video stepenu. Na ključne pita-



Procesori DX2, sa internim taktom na 66 MHz, postižu na ovom testu najbolji rezultat



Sa novom generacijom grafičkih kartica gotovo da i nije moguć rad u režimima „minimalna rezolucija/maksimalan broj boja"



Micro'486/50 MHz: *Tower* kućište sa prostorom za šest jedinica 5,25 ili 3,5 inča i napajanjem snage 230 W. Matična ploča *no name* tipa sa dve VESA 32-bitne lokal bas sabirnice, 256 K sekundarnog keša i 8 MB RAM-a, proširljivo do 32 MB i OPTi kontrolna logika. VESA lokal bas video kartica tipa StealthPro (S3) sa 1 MB RAM-a, VESA lokal bas IDE disk kontroler. Tvrdi disk WD Caviar 1250 kapaciteta 125 MB. Kolor monitor sa ravnim ekranom Eizo FlexScan F550i od 17 inča. **Namena:** grafička radna stanica za CAD i grafički dizajn. **Ocena:** procesor se sarađuje sa memorijom najbolje, ali je hardver sasvim primeren osnovnoj nameni mašine. **Korisna adresa:** 29. novembra 71, Beograd, tel. 343-940, 343-284



Comtrad'486/50 MHz: *Tower* kućište sa točkovima, zaštitnim vratima, prostorom za četiri jedinice 5,25 ili 3,5 inča i napajanjem snage 230 W. Matična ploča *no name* tipa sa dve VESA 32-bitne lokal bas sabirnice, 256 K sekundarnog keša i 8 MB RAM-a, proširljivo do 32 MB, i SIS kontrolnom logikom. VESA lokal bas video kartica tipa HG426AV sa 1 MB RAM-a, VESA lokal bas IDE disk kontroler. Tvrdi disk Maxtor 7213at kapaciteta 210. Kolor monitor Daewoo od 14 inča. **Namena:** opšta primena i server za veliku mrežu. **Ocena:** dobro uravnotežena VESA mašina bez izrazitih jakih ili slabih mesta. **Korisna adresa:** *Genex apartmani, V. Popovića 6, Beograd, tel. 222-41-81, 222-38-52, fax. 222-41-39.*



INFO-D i486/50 MHz: *Tower* kućište sa prostorom za šest jedinica 5,25 ili 3,5 inča i napajanjem snage 230 W. Matična ploča *no name* tipa sa podnožjem za Weitek 4167 koprocesor, do 256 K sekundarnog keša i 8 MB RAM-a, proširljivo do 32 MB i SIS kontrolnom logikom. ISA video kartica tipa Trident 8900 sa 1 MB RAM-a, IDE disk kontroler. Tvrdi disk WD Caviar 2200 kapaciteta 210 MB. Kolor monitor ProVitek od 14 inča. **Namena:** univerzalna primena. **Ocena:** klasična ISA arhitektura daje odličan ukupni rezultat čak i kada je video stepen izrazito arhaičan i slab. **Korisna adresa:** *Gradski park 2,8 (Zgrada Crvenog krsta), 11080 Zemun, tel./fax. 613-822.*

nje „koliko nove arhitektura personalnog računara popravlja performanse grafičkog sistema" sa zadovoljstvom odgovaramo — spektakularno. U elementarnim video testovima koji su namenjeni za upitivanje grafičkih adaptera i koji mahom govore o kvalitetu saradnje procesor/grafička kartica, nije bilo neprijatnih iznenađenja. Na novoj VESA sabirnici, zahvaljujući njenom direktnom priključku sa magistralu procesora, podaci putuju brzinom koja je 2,5 do 3,5 puta veća od brzine na ISA basu. Ovde, naravno, nije bez uticaja ni arhitektura grafičke kartice – StealthPro je nešto samostalnija i nepristupačnija od Cirrus kartice.

Oba grafička akceleratora, i mach32 i Stealth24 su, zahvaljujući arhitekturi ove klase kartica, primetno zaostale u odnosu na klasični ET4000 (i to u sporijoj mašini). ISA *Windows* akceleratori po pravilu rade pod DOS-om ispod svakih očekivanja i to zahvaljujući izuzetno sporom saobraćaju na relaciji procesor/video memorija. Stvari naravno dolaze na svoje tek kada se ove kartice pokrenu preko odgovarajućih drajvera – direktna komunikacija daje vrlo sumnjive rezultate.

Doprinos nove arhitekture podsistemu diskova nije ni iz daleka toliko spektakularan kao kod video, prvenstveno zbog toga što je brzina ovog sklopa u računaru uglavnom određena fizičkim osobinama ugrađenih diskova. Naši testovi su pokazali da su diskovi na VESA sabirnici samo 24% do 40% brži od diskova na ISA sabirnici i da ovom dobitku podjednako doprinose brzina procesora kao i propusna moć sabirnice – računar INFO-D sa kontrolerom na ISA basu je pokazao iste rezultate kao i MP Biro sa kontrolerom na VESA basu.

Jedna *mašina* (MP Biro) dolazi sa kešom od 4 MB na kontroleru. Test sa uključenim kešom je dao neverovatan rezultat – 102 disk harmonika što je gotovo udvostručilo rejting mašine. Sličan rezultat (zapravo, za nijansu bolji) smo postigli i sa *Smartdrive* kešom pa ovu „akcesnu" stanicu nismo ni uzeli u razmatranje. Ipak, vredi zabeležiti da je jedna od *mašina* stigla i u takvoj opremi.

NA STAKLENIM NOGAMA

Rejting mašina, u DOS aplikacijama, sabran je u ocenu DosMark. Ove ocene su, naravno, relativne a izvedene su tako da najpribližnije pokažu aplikativnu vrednost *mašina*. Bez obzira na čak dramatične razlike u pojedinim podsistemima, *mašine* iste klase imaju gotovo isti DosMark skor. Za nas je veliko iznenađenje predstavljala činjenica da klasična 486/50 MHz sa ISA video karticom i ISA kontrolerom tvrdog diska stoji rame uz rame sa VESA mašinama. Zapravo, ako pažljivo pogledate naše dijagrame, videćete da je za nijansu i bolja. Drugi paradoks je da je DX2/66 sa VESA video karticom postigla slabiji rezultat od DX2/66 sa ISA grafičkim akceleratorom pa čak i od jedne sasvim klasične 486/33 mašine.

To nas upućuje na zaključak da prva generacija VESA matičnih ploča i grafičkih i video kontrolera još uvek ne stoji na čvrstim nogama, ili da bi *PC Magazine* trebalo da prepen-denše svoje testove tako da realnije iskazuju ulogu nove arhitekture. Mi bismo pre bili skloni da poverujemo u ovo prvo. Tokom ispitivanja mašina suočavali smo se sa problemima tipičnim za nedorađene proizvode. Na Cirrus video karticama, i to u svim mašinama u koje je bila ugrađena, slika se kvarila za vreme intenzivnog rada diska, dok su pokreti mišom razmazivali sliku u maniru najotkačenijih ekspresionista. Iz ove svađe sa ostatkom sistema karticu je mogao da povrati samo reset mašine. Ovu vrstu problema smo primetili samo prilikom rada testa *PC Benchmark*, ali se i u drugim aplikacijama, ali to može samo da znači da nismo pustili dovoljno drugih programa. Diamond StealthPro je kvario sliku u rezolucijama sa velikim brojem boja (na primer

Nastavlja se na strani 25



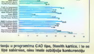
Iz memorijskih testova se mogu izvući najmerodavniji zaključci o kvalitetu dizajna matičnih ploča



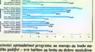
Dobici od 2,5 do 3,5 puta (reper ET4000) u brzini video podsistema *jesu* spektakularni



Dobici od 25 do 40% u komunikaciji sa tvrdim diskom nisu spektakularni ali ni za – potcenjivanje



U crtanju u programima CAD tipa, Stealth kartice, i to za oba tipa sabirnice, nisu imale ozbiljniju konkurenciju



Korisnici *spreadsheet* programa se moraju da budu naročito pažljivi – sve kartice sa testu su dobro opsluživale ovu vrstu softvera



Uz brzinu u programima za slikanje i obradu fotografija u boji, kartica mach32 u pogodnostima ide korak ispred konkurentskih proizvoda na tržištu

UPRKOS SVIMA, UPRKOS SVEMU

Međunarodni sajam tehnike i tehničkih dostignuća, koji se od 10. do 15. maja ove godine događao na Beogradskom sajmu, pod geslom „korak u budućnost", po 37. put, bio je prilika da se domaće firme i korisnici računara okupe i – uprkos embargu i sankcijama – vide, razmene ideje, prezentiraju softver, hardver, te nove/stare projekte i, pre (ili nakon) svega, obrnu novac.

Iako bi neki računarski cinik verovatno primetio da su zauvek prošla vremena velikih sajamskih uzbuđenja, 350 izlagača (od kojih 90 u već legendarnoj „računarskoj" hali XIV) na ovogodišnjim majskim danima tehnike koji su direktno na svojoj koži osetili damoklov mač Rezolucije 820, teško da bi se složilo sa njim „Neposredno posle izglasavanja rezolucije imali smo desetine telefonskih poziva Mišljenje je bilo da bi se moglo i odustati od sajamskog nastupa U veoma loš čas pale su i one prethodne sankcije dok su ove najnovije u jednom momentu čak ugrozile održavanje sajamske manifestacije Srećom, do toga nije došlo " – kaže **dr Siniša Zarić** generalni direktor Beogradskog sajma

Vesna Ćosić

POSLOVNA ATMOSFERA

Pod geslom „korak u budućnost", na dvadeset hiljada kvadratnih metara izložbenog sajamskog prostora – posetioci su mogli da vide najnovija dostignuća iz elektronike informatike elektrotehnike telekomunikacija, biotehnologije energotehnologije, ekologije, građevinarstva – iako zbog embarga i sankcija međunarodne zajednice prema SRJ nije bilo direktnog učešća izlagača iz inostranstva „Ali ovde se zato našao veliki broj proizvoda stranih firmi posebno u oblasti informatike koje su posredno prisutne preko svojih zastupnika i drugih oblika međunarodne trgovinske saradnje Našao se način da se i njihovi proizvodi vide na sajmu da se obavi trgovina i da sajam zaista predstavlja jedno mesto susreta." – kaže gospodin **Zarić**

I zaista, kada preživite gužvu na ulazu (dva portira mala vratanca), prodavce ušećerenog voća žute traktore i ćevabdžije i dokopate se hale XIV, *computer show* na beogradski način već s vrata je poručivao – nema da nema! (Skoro) svi su tu! IMTEL, ADACOM, BS PROCESOR GAMA, ADA COMPUTERS CET PUPIN EI BULL COMTRAD DIGIT ELECTRONIC DESIGN, GLOBAL, IBIS SYS JUGODATA MIMICO MIKRONOVA MICROSYS MISOFT, NUKLEUS PC PLUS PERI HARD, PIXEL, SAGA INFO-D i još 66 računarskih firmi – učinili su da se i u ova oskudna vremena na jednom mestu zavrtimo u računarskom ritmu To što su u međuvremenu veliki postali manji, a mali veliki dalo je naslutiti „šta bi bilo kao bi bilo" – u vreme bez sankcija, embarga, ljutih monopola, državnih protektorata – i ilegalnog biznisa kao ilegalnog građanina Stvarnosti „Proizvodno preduzetništvo, pokret koji se već nekoliko godina pomalja zajedno sa prestrukturiranjem privrede, učinilo je da naše tržište bude bogatije za niz proizvoda Za nas kao upravu Beogradskog sajma veoma je važno i vredi istaći da preduzeća sve više vladaju tehnikom marketinškog nastupa Moderni senzibilitet novih poslovnih ljudi ogledao se na ovogodišnjem Beogradskom sajmu Za razliku od nekadašnjih petrificiranih sajmova koji su bili fotografija tadašnje privredne strukture u kojoj su preovladavali veliki zavodi i kombinati, pokazali su daleko više raznovrsnosti, boje, inovacija, roba, poslovne atmosfere " – kaže **dr Siniša Zarić**, generalni direktor Beogradskog sajma

To što su računardžije dale sve od sebe dizajnirajući sasvim profi, izložbeni prostor učinilo je da na prvi pogled ne upada u oči da na galeriji nema izlagača da su pojedine „korisna površina" prazna da mnogi ljudi od računarskog rejtinga više nisu u „domaćem filmu" Jedni su svoje štandove presvukli u ljubičasto drugi su promenili logo treći su obesili dinosaurovsku telefonsku slušalicu, četvrti su se transformisali u šarene bašte peti se ufurali u *light show* šesti su akcentirali klasični šank. I tako redom, od štanda do štanda, bum dizajnerskih caka i lepih devojaka (čiji bi profesionalni osmeh svako popodne obično iskopneo, da bi sutradan ujutro bio opet kao nov) Ali ako daće Bog, doći će vreme kada ćemo biti u stanju da se smeškamo po vasceli dan

EKSPLOZIJA BOJA I ZVUKA

Stariji posetioci ako nisu bili u poslovno-partnerskom fazonu i tvrdili pazar iza paravana, oprezno su zagledali izloženi hardver, radoznalo obilazili štandove multimedijalne provenijencije, sećajući se, valjda daleke 38-me godine kada je sa Beogradskog sajmišta iz Philips-ovog paviljona emitovana prva televizijska slika na Balkanu Klinci su cepali igrice najviše kod IMTEL-a i ADACOM-a sedeći poput voštanih figura i igrajući pre svega KEEN 1, 2 3 4 5 6, 7 (ovogodišnji hit nad hitovima među osnovcima) IMTEL se za ovu priliku bacio u pop art „piramida" imidž asocirajući „one između" na legendarno Commodoreovo egipatsko selo koje je daleke 88 godine ovaj proizvođač instalirao na Personal Computer Show u Londonu koji je tada bio udaljen taman toliko koliko traje let aviona. Bilo kako bilo muzika je treštala na sve strane a CD ROM- ovi I SOUND BLASTERI ušetali su u svakodnevni računarski život. No da računarski narod nije samo u fazonu da udari „brigu na veselje" okupiran muzičkom računarskom podrškom video se i po gužvi na štandu ADACOM-a veliki interes kolega i posetilaca izazvala je prezentacija programa MAGIC i izlaganje prof dr Milana Kukrike

LET IT BE – pevali su Bitlsi u kompjuterskom izdanju na štandu SAGE, čiji je disko dizajn bio zanimljivo okruženje za izloženi hardver Gospodin **Goran Đaković** direktor preduzeća Saga šapnuo mi je na uvo (jer se drukčije nismo mogli čuti) sledeće

„Prisustvujemo na Sajmu uprkos ovim teškim uslovima, uprkos embargu i sankcijama. Zadovoljan sam Mi smo ove godine pošto se naša pozicija iskristalisala kao pozicija sistem integratora izložili više malo teže opreme jači računari integracija tih računara u mrežne si-



Ekipa časopisa „Računari": Katarina Milovanović (lice sa naslovne strane), Vesna Jeremić (komercijalni urednik), Ranka Jovanović (prevodilac), Irena Durmišević (sekretarica „Somome") i Jovan Regasek (glavni urednik)



Saga je izložila računarsku opremu za najteže poslove Goran Đaković, direktor preduzeća „Saga" u društvu sa gradonačelnicom Beograda dr Slobodankom Gruden



Kako smo se nadali - dobro smo se udali. Pogled na halu XIV ribljim okom iz ptičje perspektive



Dr Siniša Zarić, direktor Sajma: Na ovogodišnjem Sajmu tehnike je došao do izražaja sasvim nov poslovni senzibilitet



Delina piramida „Imtelovi" inženjeri su neumorno demonstrirali posetiocima prednosti „Imtelovih" softverskih i hardverskih rešenja

steme Novell i Unix sa SPARC procesorima, sa SUN-ovim serverima, grafičkim stanicama. Izložili smo i naš distributivni program japanske Star štampače, američke (Athene) magnetna medija, kao i American Power Conversion – backup and smart UPS-ove. Štand je dizajnirao gospodin Boris Miščević. Posaban trud je uložen iz razloga što smo promenili vizuelni identitet firme."

NAJVAŽNIJI SU LJUDI

Sajam ne bi bio sajam bez pratećih poslovno-stručnih skupova, čija posećenost ove godine nije omanula. Naučno-stručni skup „INFORMATIKA" u organizaciji Društva za informatiku Srbije (Informacioni sistemi, Razvoj baza podataka, Savremene telekomunikacije, Novosti u razvoju hardvera i softvera), kao i „MULTIMEDIJALNE INFORMACIONE TEHNOLOGIJE" – bili su svakako najzanimljiviji oko prateće „računardžije". No, „programiranje i informatika" bili su predmet i drugih specijalizovanih skupova koji se ovde održavaju. Recimo, na Sajmu građevinarstva koji se odvijao u istom periodu kada i Sajam tehnika, kada se govorilo o „Upravljanju građevinskim projektima" – očito se puno insistiralo na dokumentacionim bazama i načinu korišćenja podataka u upravljenju projektima u građevinarstvu. Tako su i ostale oblasti imale računarstvo kao potporu." – rekao mi je gospodin **Zarić**. Na njegovu sugestiju, otišla sam i na drugi etaž Hale IV. Razlog je dobar i toplo se preporučuje, jer tu se nalazi Svetski trgovački centar, formiran pre nepunih godinu dana. Veze i komunikacije funkcionišu preko satelita, vrši se normalna distribucija elektronske pošte, postoji kontakt sa pet miliona preduzeća u celom svetu, a raspoložive baze podataka se koriste na dnevnoj osnovi.

No, vratimo se „računarskoj" hali XIV. Ako me je išta impresioniralo, onda je to komunikativnost i međusobne interakcije ljudi po računarskim firmama. Prosto, neko novo doba je kucalo je (ipak) i na naša klimave vrata, jer svima je prosto sasvim familijarno mišljenje da bez međusobne saradnje, bez obzira na konkurentnost, nema ništa od profesionalnog nastupa. Jednostavno, onaj ko još nije shvatio značaj razmena informacija i zajedničkog delovanja, kreiranje tržišta, osuđen je na getoizaciju. Iz tog ugla, potpisnica ovih redova se na momente osećala kao nekad davno u Dubrovniku, gde bi se preko leta „preselio" ceo Beograd. Ne sve strane poznate face ljudi koje nameie prilike da sretnete preko godine oko štandova kolege i ljuti konkurenti ugodno ćaskanju, prosto – druže se! Recimo, na izložbenom prostoru COMTRADA, koji se kupao u svojim tradicionalnim bojama (žuto, crveno, plavo, zeleno), videla sam se i sa **Petrom Damnjanovićem** iz OSE, **Borom Hadžibabićem** iz SEL-a, **Milanom Stevanovićem** iz PIXEL-a. To što su im interesi konkurentni ni najmanje ih nije omelo da imaju dobru međuljudsku i profesionalnu komunikaciju. Oprostite što ovoliko insistiram na tome, ali u doba kada preovlađuje trend uvlačenja u vlastite školjke, kada je izolacija i stanje i način mišljenja – lakoća komunikacije računarskih biznismena je neverovatno lepa i čemu kriti, budi nadu.

Sok sam popila na štandu firme **Info-D** ugodno razgovarajući sa gospodinom **Draganom Cvetićem** koji mi je ispričao sledeće: „Imamo izuzetan kvalitet opreme. Svaka naša komponenta je jedna od najpoznatijih u svetu. Uveli smo i neke novine. Doneli smo posebne radne stanica sa Tajvana. To je kompletan računar u kućištu tastature. Služi i kao radna stanica pod bilo kojim operativnim sistemom, DOS-om, Novell-om, Unix-om. Ove godine krenuli smo jače sa lap-top-ovima i note-book-ovima. Doneli smo veliku količinu kolor lap-topove počev od 386sx preko 486sx-a do najnovije verzije koja je 486DX2. To je praktično mašina koja u standardnoj konfiguraciji teži oko 60-70 kilograma – a kod nas oko pet kilograma. Četvrtu godinu smo na sajmu. U odnosu na ostale sajmove, moj utisak je da sajam nije ispunio ni jednu trećinu očekivanja. Ipak zadovoljan sam izgledom i ponudom. Ali, to nije promet kakav je bio zadnjih dve tri godine. Mislim da je kompletno sve opalo za nekih šezdeset posto. Vrlo je mali broj pojedinačnih privatnih korisnika, dolaze nam oglavnom firme, jer iako su cene pale na našem tržištu, ipak je to strahovito visoko za prosečnog privatnog kupca. U saradnji sa Genex-om napravili smo softver koji je počeo već da se vrti u svim Genex-ovim predstavništvima u svetu. Uredili smo i jednu verziju korisnog alata za *Clipper* programere. To je takozvana Info-D Clipper biblioteka koja omogućuje da korisnik više ne mora da gubi dane i dane na kreiranju nekih programa."

I DALJE PUNOM PAROM

To što na sajmu nije bilo nekih firmi, čije je prisustvo na domaćoj računarskoj sceni vidno, nije isključilo i prisutvo ljudi iz njih. Premišljajući se između izbora ovog ili onog „slučajnog prolaznika", opredelila sam se za gospodina **Petra Damjanovića**, jednog od vlasnika OSE: „Ove godine smo odlučili da ne izlažemo. Razlog tome je opšta situacija na tržištu. Obzirom da smo mi firma specijalizovana za profesio-

nalna grafičke periferijska uređaje naša procena o toma kako će izgledati ovaj sajam i naša procena izrišla opradelili su nas da se ovoga puta odlučimo na – na učestvovanje Kao posetilac do sada nisam video ništa značajno novo što se naravno moglo i očekivati To je uglavnom roba koju poznajemo po cenama koje su sve niže i niže verovatno ispod razumnih granica rekao bih Drago mi je da vidim da kolege rade dobro Da ima firmi koje se i dalje razvijaju i koje mogu da opstaju u ovim teškim uslovima Ali sve u svemu – daleko ispod ranijih godina"

Gospodin **Milan Stevanović** iz firme PIXEL (firma poznata i po toma što jedina u YU ima Silicon Graphics grafičku radnu stanicu) – misli nešto drugačije „Što se tiče samog sajma mi smo izuzetno zadovoljni Napravili smo i specijalni sajamski popust – 10% od naših standardnih cena stalno se trudimo da na neki način spustimo cene Mislimo da više nema potrebe kupovati računarsku opremu u inostranstvu jer smo sada jeftiniji nego u Nemačkoj To smo uspeli zahvaljujući privatnim firmama koje su u zadnje dve godine počele da kupuju robu naveliko omogućavajući našem kupcu da nabavi računar po sasvim solidnoj ceni Ova godina sajam deluje mnogo opskurnije u smislu propagande ljudi koji izlažu broj firmi koja su došle Prošle godina cela galerija je bila puna Međutim poseta je izuzetna Ja, naravno govorim iz projekcija mog standa Svi koji dođu imaju nameru da kupe Ima malo šetača Mi smo se pripremili za sajam Napravili smo i oba spota – za Sajam tehnike i za Sajam građevinarstva Svi su zadovoljni mi smo zadovoljni Upravo smo kupili još dve radne stanice jače od one koju već imamo Planiramo da podignemo nivo kompjuterske grafičke animacije da sada izađemo na inostrano tržište i da jednostavno, a našim cenama sa poznatim srpskim znanjem i umećem – a mi to stvarno imamo kao što smo i u košarci uspešni – uspemo Ukratko veoma perspektivno i ja se tako osećam "

MOŽDA I U SEPTEMBRU

Sajamski optimizam nije zaobišao ni ljude iz COMTRAD a, firma čiji je rejting veoma visok među kolegama iz računarskog biznisa Na više mesta sam čula da računare prodaju u ogromnim količinama pre svega zahvaljujući vrhunskim stručnjacima za plasman a kada to kaže konkurencija (po sistemu „skidamo im kapu") onda valja verovati A gospodin **Veselin Jevrosimović** iz KOMTRAD-a kaže Sajam je ove godina bio stvarno izuzetno posećen Van očekivanja! Ove godina na štandu smo izložili pored kompletna linije naših računarskih proizvoda i Canon fotokopir mašine Casio kalkulatora i Hewlett Packard laserske štampače " Na COMTRAD-ovom standu bio je prisutan i radio PINGVIN S razlogom Dizajnirali su softver koji se „ubio" za korišćenja u radijske multimedijalna svrhe što će reći prava stvar za profesionalni radio Tonac praktično uopšte više nije potreban Izbacuje se korišćenje REVOX a magnetofona kasetofona Sve se dobija pritiskom na samo jedan taster Pristup je 14 mili sekundi Sve u svemu soft sound tim koji vodi **Aleksandar Šušnjar** uradio je dobar posao Nisla čudno onda što je multimedija zauzela centralno mesto na ovogodišnjem sajmu kao što je kako kaže gospodin **Veselin Jevrosimović** iz COMTRAD-a – zauzimala centralno mesto i na ovogodišnjem svetskom sajmu elektronike u Hanoveru

„Danas kada razgovaramo sa izlagačima oni smatraju da bi Beogradski sajam trebao da bude dva puta godišnje domaćin takvog tipa sajamske prezentacije Postoji ideja da u jesenjem terminu bude Informatička biro oprema Neka vrsta Biro-expo-a Iz ankete izlagača proizilazi da oni vole ovaj sajam koji je velik koji im obezbeđuje medijsku podršku i publiku ali da bi još neki takav program imao punog smisla

Nama je puno srce – ne zbog veličine prodatog prostora niti zato što smo uprkos sankcijama ostvarili jednu ovakvu priredbu – već zbog toga što vidimo da su naša nova preduzeća a i ona stara – ovladala marketingom i znaju kako da iskoriste sajam da bude u funkciji unapređenja prodaje " – rekao mi je gospodin **Zarić** generalni direktor Sajma

Prema tome doviđenja u septembru!

IPAK SE KREĆE

Najveći računarski događaj godine, Beogradski sajam tehnike, je za nama Utisci su različiti Jedni ga kude jer su, kako kažu, devojke na standovima bile ružnije nego proteklih godina, drugi se bune jer je muzika sa pojedinih štandova bila preglasna, treći su pak zadovoljni zbog lepog vremena tokom sajma Ne isključujući značaj navedenih primedbi, ostaviću ih po strani i koncentrisati se na izložene računare i računarske proizvode

Na najvećem broju štandova kao što se i očekivalo bili su izloženi PC računari Za razliku od prošlogodišnje izložbe 286 ploče kojih je istine bilo ali veoma malo odlaze sa scene Većina prodavaca nude 386SX ploče kao deo osnovne konfiguracije Takođe prošlogodišnje zvezde štandova, ISA i EISA 486-ce prepustile su počasno mesto 486 pločama sa *local bus*-om i *overdive* procesorom Ljudi iz „Ada Computers" a su sa ponosom pokazivali svoju mašinu sa VESA *local bus* 486DX pločom na 33 MHz sa *overdrive* procesorom i internim taktom od 66 MHz I jednom zanimljivošću – podnožjem za Pentium novi Intelov procesor koji po inerciji mnogi nazivaju 586

VESA *local bus* ploče su bile ukrašene i odgovarajućim *local-bus* video karticama koje su plenile mnoštvom boja i pre svega brzinom *Windows* aplikacija atraktivne animacije i još atraktivnije digitalizovane slike lepotica demostrirali su mogućnost novih kartica Ipak sve ovo nije moglo prikriti činjenicu da su ostali *local bus* slotovi (uglavnom ih ima dve) umesto da nose *local bus* disk kontrolere na većini računara bili prazni

VGA video kartice su i sa najjevtinijih mašina potisnule Hercules kartice te osnovni video sistem sada čine VGA kartica sa odgovarajućim VGA mono monitorom A monitora je bilo za svačiji ukus iako nismo primetili neki veliki pomak u odnosu na prošlogodišnju izložbu – osim da Hercules monitori nestaju sa Hercules karticama kao i da su ove godine izlagači prikazali mnogo više monitora sa dijagonalom od 17 inch a i većom Na žalost kako su mnogi primetili na Sajmu se nije mogao naći kvalitetan filter Izloženi filteri su uglavnom bili najjevtiniji modeli svetskih proizvođača

Prenosni računari su krasili većinu štandova kako *notebook* tako i *subnotebook* *Notebook* računari bez problema nose u sebi 486DX procesore i kolor ekrane Kod INFO D e smo videli 486DX/2 *notebook* na 50Mhz Zanimljivo je bilo posmatrati *subnotebook* na „Imtel" ovom štandu koji se lepo snalazio sa mrežnim softverom i *Windows*-ima dok su mrežna karta i miš bili gotovo veći od samog računara „LS Dele" je predstavile i novu seriju *Psion* džepnih računara od kojih je svakako najinteresantniji *Psion Series 3* o kojem ste imali prilike da čitate na stranicama našeg lista

Kao što se i očekivalo računarske komunikacije i mreže su zauzele istaknuto mesto Gotovo da nije bilo izlagača čiji računari nisu bili vezani u lokalnu mrežu Najviše je svakako bilo *Novell* mreža premda se moglo primetiti i prodor *Unix* mrežnog softvera (NFS TCP/IP) kao i povezivanje mašina pod različitim operativnim sistemima u jednu mrežu Najdalje je otišao „Imtel" koji je na svom štandu imao mrežu u kojoj su bile mašine pod MS DOS om *Unix*-om i *Windows NT* om Na „Imtel"-ovom štandu su svakodnevno skidali i *Public Domain* programe sa *Astra* satelita. Naime satelitski program Pro-Z celog dana emituje PD programe paralelno sa video signalom Uz pomoć posebnog uređaja i softvera dok vi gledate satelitski program vaš disk se puni PD programima Kao trenutno jedina računarska veza koja još uvek funkcioniše sa svetom JUPAK je dobio mesto na standovima nekoliko izlagača Naravno tu su bile i odgovarajuće kartice i softver za priključenje na ovu mrežu Pored ovih kartica očekivao sam da među izloženom opremom modemi i fax/modem kartice zauzmu značajno mesto zbog sve veće popularnosti modemskih komunikacija međutim prevaren sam sa Istina ovih uređaja je bilo ali je ponuda bila zaista skromna

Iako su PC računari sa MS DOS om i *Windows*-ima vladali ovom izložbom bili su zastupljeni i drugi operativni sistemi Pre svega *Unix* a zahvaljujući „Digit" u VMS te kod „Imtel" a *Windows NT* Čudno je što se nigde nije mogao videti OS/2 *Unix* na PC računarima je uglavnom predstavljao *SCO Unix* dok su na „velikim mašinama tu bili *Ultrix* (verzija za „Digital" ove mašine) i SUN OS Ovaj poslednji je bio na Saginom štandu i vrteo se na *SUN SparcStation 2* grafičkoj radnoj stanici zakačenoj na *SparcServer* sa dva RISC procesora impresivnih 256M RAM a 3G diska i još impresivnijom cenom od 70 000 dolara *Ultrix* je radio na *Digital* mašinama na štandu „Digit" a gde su se mogli videti i VAX-ovi 4000 200 i 4000-300 DEC system 5400 *VAX station* i *DEC station* kao i odgovarajuća oprema za ove računare

Prateće opreme za računare je bilo ali opšti je utisak ipak nedovoljno Jedine svetla tačke su trake za matrične štampače Dva izlagača su na svojim štandovima imali isključivo ove proizvode a to su „Proteus" i HBM Ove firme kao i „Penhard" su nudili i zamenu trake u starim kasetama novim Iako već viđenu zanimljivo je bilo opet videti multimediju na delu Pažnju posetilaca pogotovu onih mlađih su privlačile razne zvučne kartice TV tjuneri za praćenje TV programa na monitoru računara džepne mašine za igru i mnogo drugih sitnica Istaknuto mesto su imale i animacije Na štandovima firmi „Gama" „Pixel" i još nekih koje nude izradu animacija videla su se dela umetnika koji rade za njih

Sa softverom je priča vrlo slična onoj prošlogodišnjoj Dakle softvera je bilo malo ili ga uopšte nije bilo na štandovima izlagača Da budem precizniji nije bilo originalnog softvera dok je piratskih kopija bilo na pretek Časni izuzeci su „El Bull" koji je predstavio *Magic* i *Wizdom* generatore aplikacija CET koji je nudio „Microsoft" ove i „Borland"-ove pakete (*Access Pascal 7 0*) „Jugodata" i još neki Jedino je domaćeg softvera bilo nešto više nego prošle godine s tim što su to uglavnom bile aplikacije za pomoć u vođenju poslovanja

Još gore je bilo sa knjigama Skroman izbor kod „Tehnikom a i nešto veći izbor ali uglavnom starijih izdanja kod CET a je sve što se moglo videti

I eto zavesa je spuštena izvođači su otišli na zaslužen odmor a gledalište je uprkos zamerkama na predstavu izašlo uglavnom zadovoljno jer – ipak se kreće

Pavle Paković

NAUKA VIŠE NE STANUJE OVDE

Dr Vlastimir Matejić, direktor Instituta „Mihajlo Pupin" i donedavno, u vladi gospodina M. Panića, ministar za nauku, tehnologiju i razvoj govori o odlivu mozgova, naučnoj politici Jugoslavije i strategiji preživljavanja.

Dr Vlastimir Matejić, elektroinženjer, kako sam kaže – „stare generacije" (diplomirao pre 37 godina), magistrirao je ekonomske nauke (u ono vreme nova disciplina operaciona istraživanja) a doktorirao na tehnici. Karijeru je počeo na Elektrotehničkom fakultetu („fizika mi je bila slabost"), odakle se „preselio" u Centar za obrazovanje rukovodećih kadrova u privredi. U Centru „otkriva" menadžment i daje sve od sebe da ova ustanova postane evropski konkurentan Centar za menadžment. No, stvari obično ne idu glatko: reforma iz 65/66. godine je to uništila, kako kaže dr Matejić, sa idejom da menadžeri postaju iz radničke klase, da samo radnička klasa može da proizvodi menadžere, da svaki čovek može da bude menadžer. Posle godine dana provedene u Americi, dr Vlastimir Matejić dolazi u Institut „Mihajlo Pupin", gde punih 14 godina razvija oblast operacionih istraživanja. Počinje da se bavi i problemom razvoja („to me i danas najviše zanima") i osniva pri institutu Centar za istraživanje razvoja nauke i tehnologije, funkcioner je u Komitetu za naučno-tehnološku politiku OECD-a, a poslednjih godina je generalni direktor Instituta „Mihajlo Pupin". U vladi gospodina M. Panića bio je ministar za nauku, tehnologiju i razvoj.

Najviše me je zanimao problem razvoja i to sa različitih kulturnih stanovišta – evropski pristup, američki pristup... Predajem na beogradskom Elektrotehničkom fakultetu „Upravljanje organizacijama", predavao sam na dosta univerziteta u zemlji i na nekoliko u inostranstvu; moja univerzitetska nastava je bila najviše iz domena operacionih istraživanja (uglavnom postdiplomska nastava). Dok sam bio Savezni ministar za nauku, tehnologiju i razvoj, zalagao sam se za tezu da razvoj u ovoj zemlji treba da bude zasnovan na najvišoj nauci i tehnologiji. Dva puta sam imao ponudu da budem ministar za nauku i tehnologiju u Republici Srbiji. Ja to nisam prihvatio samo iz jednog razloga – ulazak u državnu administraciju udaljio bi me od istraživanja. U ovu vladu sam ušao sa dogovorom da će to biti samo do kraja prošle godine i sa zadatkom da organizujem Ministarstvo, pokušavajući da nekim svojim vezama u inostranstvu, pre svega preko OECD-a i Komisije Evropske zajednice, delujem na smenjenje sankcija za nauku. Ministarstvo sam organizovao, sa sankcijama ništa pozitivno nisam učinio.

Imali ste priliku da vidite, učestvujete i direktno osetite sve naše uspone i padove. Da li ste ikada kao naučno istraživački radnik, bar intimno, poželeli da odete iz zemlje?

Ja tu želju intimno oblikovanu – ne govorim o željama proisteklim iz ljutnje – nisam imao. Verovatno iz razloga sasvim lične prirode. Počev od moje relativno jake vezanosti za ovdašnju kulturu, prijatelje, roditelje, rodjake, za podneblje ako hoćete. To tumačim izvesnom vrstom kulturnog otpora. Imao bih uverenje da sam izgubio nešto od ove kulture u koju ubrajam sve – i pozorište i literaturu, i sport, i klub za koji navijam, prijatelje koje imam, roditelje.

Međutim, odlazaka je uvek bilo. Moram da vas pitam kako doživljavate odlaske ljudi iz vaše kuće, iz Instituta „Mihajlo Pupin".

Da se razumemo, kada su u pitanju naučni radnici iz Jugoslavije, odlazaka će uvek biti više nego dolazaka. Mi smo zemlja nato emigracije. Što je potpuno prirodno. Odlaziće ljudi iz ove zemlje – i u tom nema ničeg lošeg sve dok broj onih koji odlaze ne postane zabrinjavajući kao što je to slučaj sada. Iz Instituta „Mihajlo Pupin" ljudi su odlazili, fluktuacija iz Instituta uvek je bila prema Univerzitetu u Beogradu, nešto manji broj prema privredi Srbije, Jugoslavije, odnosno u druge institute i – postojao je uvek odlazak u inostranstvo. Mi smo kuća kroz koju jedan deo ljudi prosto prođe. Nauči se kod nas, stek-

Ako bi se desilo da ove godine ode još dve hiljade ljudi, onda iduće godine, 1994, neće imati ko da ode!

Vesna Ćosić

ne kompetentnost i konkurentnost i ode. Mi smo to amatirali svojom komparativnom prednošću.

Ne, sadašnji je odliv preveliki! Prošla godina je otišlo preko sedamdeset ljudi. Odlasci se i ove godine nastavljaju. Već u prvih tri meseca otišao je veliki broj ljudi. A prema onome što znamo, veliki broj ljudi ima rastuću sklonost i spremnost da ode.

Koje su neposredne posledice odlaska stručnih ljudi?

U prvih nekoliko meseci ove godine imamo čak i neočekivane pojave. Radi se o tome da je odlazak pre uglavnom bio vezan za ljude do tridesetih i pete godine, sada se dešava da se čovek u pedesetim godinama reši da potpuno promeni okruženje. Sadašnji odliv vrlo pogađa Institut, pogađa ga više nego druge institute. Ne samo zato što je mnogo ljudi otišlo i što i dalje odlazi, nego zato što smo mi institut koji još uvek ima projekte. Znate, sada kad ljudi odu, mi ne kažemo – neko je otišao, hvala Bogu, manji problem, nego – otišao je i napravio nam problem. Jer, ne možemo da završimo projekat na vreme, ne možemo da završimo po istoj ceni itd.

Ove sankcije su nas stavile u situaciju koja je metaforično slična zatvoru. Šta onda da radimo? Jedna varijanta je da u zatvoru igramo karte. Druga varijanta je da učimo. Da ne budemo slepci!

Prijem novih ljudi postoji – eto, juče sam primio inženjera na posao, ali to su mladi ljudi kojima treba tri do četiri godine rada da mogu da zamene one koji su otišli. Prema tome, vrlo nas pogađa. Za institut koji se finansira iz fondova, koji je prijavio sto naučnih radnika, odlazak ljudi može da olakše finansijsku situaciju. Ali to je kratkoročna viđenja stvari.

Čula sam od jednog kolege koji je bio u Kanadskoj ambasadi sledeću stvar koja zvuči kao crni humor. Naime, službenik u imigracionom odseku pitao je čoveka koji je tražio ulaznu vizu za Kanadu: „Izvinite, samo da vas pitam – da li u Institutu „Mihajlo Pupin" još uvek ima ljudi?" Kanadski službenik je to rekao bez namere da bude duhovit ili ciničan, jednostavno je samo bio radoznao. To zvuči strašno surovo.

Saznao sam sasvim pouzdano od novinarke iz jedne kanadske agencije da joj je u Ambasadi rečeno da je u ovoj godini udvostručen interes za odlaskom u Kanadu. Naravno, oni se mogu pitati ko to još odlazi! Ima li uopšte više ljudi!? Prema mojoj proceni, prošle godine je otišlo oko hiljadu ljudi koji se bave istraživanjem u prirodnim, tehničkim i tehnološkim naukama. Svako ko je otišao prošle godine iod će, automatski, govoreći, sklonosti kod dvojice novih da odu ove godine. Ovo što sam saznao iz Kanadske ambasade potvrđuje moja nagađanja šta to se događa. Udvostručio se broj zahteva za odlaskom. Ako bi se desilo da ove godine ode još dve hiljade ljudi iz ovih oblasti istraživanja, a ja računam da u Srbiji ima oko pet šest hiljada ljudi koji se bave istraživanjem i razvojem u tehničkim naukama – onda iduće godine 1994. nema više ko da ode!

Iz te reakcije službenika Kanadske ambasade ja zaključujem da je verovatno većina postojećih istraživača u Institutu već bila u Kanadskoj ambasadi! Već je pokupila formulare, upisala se i čeka.

Međutim, bez obzira na sve, na situaciju i okolnosti u kojima se nalazimo, vi ste, kao savezni ministar za nauku, uradili projekat Politike naučno-tehnološkog razvoja SR Jugoslavije, vidim štampan devedeset treće godine, da li još uvek važi, s obzirom da ste se vratili u Institut i da više niste savezni ministar?

Uspeo sam, za vreme dok sam radio u Vladi, da napravim tu politiku, koja je bila na vrlo obimnoj i intenzivnoj javnoj raspravi u naučno istraživačkom sistemu cele Jugoslavije, u svim univerzitetskim gradovima u Srbiji, Crnoj Gori, u Srpskoj akademiji nauka i umetnosti, u Crnogorskoj i tako dalje, uspeo sam da je proguram tako da je Savezna vlada tu politiku usvojila i ona je objavljena u Službenom listu SR Jugoslavije. Ona je obavezujuća politika za vladu SR Jugoslavije. Obavezujuća u smislu značenja reči „obaveza" u civilizovanom svetu. Za realizaciju te politike potrebni su uslovi, pre svega legislativne prirode. Dok sam bio tamo, uspeli smo da uradimo predlog Zakona o finansiranju naučno istraživačkog razvoja SR Jugoslavije. On je podnesen Skupštini još u prošlom sazivu. Ona je to, kao i sve zakone, praktično odložila, nova Skupština ga je zadržala u proceduri i mislim da će biti usvojen. To je sjajna stvar.

Na sednici Savezne vlade od 11. februara 93. donete je rezolucija o politici naučno-tehnološkog razvoja Jugoslavije sa deset primarnih ciljeva. Koliko tih deset primarnih ciljeva mogu da se realizuju s obzirom na realne okolnosti i ovaj „Berlinski zid" koji nas je okružio do neba?

Prvo, politika naučno-tehnološkog razvoja SRJ je pravljena za specifičnosti Jugoslavije, ali nije pravljena tako da reši problem razvoja u uslovima sankcija. Iz prostog razloga što u uslovima sankcija i nema razvoja. Apsolutna je zabluda, neozbiljnost, površnost – da neko tvrdi da dok traju sankcije mi možemo da činimo nešto na razvoju. Dok su sankcije, mi možemo samo da pokušavamo da na nižem i nižem nivou aspiracija preživimo. Ta zabluda je vrlo opasna, jer ona produžava spremnost za sankcije.

Realizacija ovih ciljeva koji su zacrtani u politici ukazuje na drugu bitnu stvar koja je cena ostajanja pod sankcijama! Ostati pod sankcijama znači ne realizovati te ciljeve, tako da se pojavljuje državni, vladin problem. Ciljevi razvoja moraju da budu na prvom mestu, jer tek kada oni budu ostvareni, moći će da se ostvaruju drugi nacionalni ciljevi političke prirode, etničke prirode i tako dalje. To je suština stvari.

Program javne naučno-tehničke i tehnološke infrastrukture predviđa, između ostalog, i centre za tehničke i tehnološke inovacije, povezivanje sa međunarodnom infrastrukturom. To zvuči samo kao lepa želja.

Vladina politika u oblasti naučno tehnološkog razvoja mogla bi i da da bude i dalje dobra, da se redukuje na problem izgradnje javne naučno istraživačke infrastrukture. Zemlja koja ima ovu infrastrukturu dobru, stavi je na raspolaganje onima kojima je potrebna, znači privredi. Oni bazirajući se na toj infrastrukturi obavljaju istraživanja, finansiraju i stvar se odvija kako treba. To je prva napomena.

Druga napomena iza ova politika, kao nešto što se može samo između redova pročitati, jeste uverenje da koncept regionalnog razvoja treba da bude vladajući razvojni koncept u Jugoslaviji. Ljudi tom konceptu prilaze opterećeni najčešće nekim dnevno političkom nerazumnostima. Neki kritičari, kada sam to saopštavao, rekli su – to mi miriše na pravljenje pokrajina. Glupost! Sve što treba „da miriše" u vezi sa regionalnim razvojem jeste da svaki region kod nas ima neke razvojne specifičnosti koje predstavljaju njegovu komparativnu prednost i naučno-tehnološki razvoj treba da ima to na umu. Zbog toga je rečeno da jedan deo naučno-istraživačke infrastrukture treba da ima regionalni karakter. To mogu biti inovacioni centri, centri za konsalting, centri za difuziju tehnologija i tako dalje.

Drugi deo vašeg pitanja odnosi se na saradnju, uključenje u evropsku ili međunarodnu naučno-istraživačku infrastrukturu. Dok je sankcija – toga nema! Međutim, na

Povratak emigranata iz populacije naučno-istraživačkih radnika je jako mali svuda u svetu

Naučno-istraživački sistem treba da sačuva svoju unutrašnju disciplinu i unutrašnju naučnu odgovornost bez obzira na uslove

biti povezan sa naučno-istraživačkom infrastrukturom izvan Jugoslavije znači – osuditi sebe na propast! To je razlog više da se izvrši revizija nikada nepisanih, ali podrazumevajućih takozvanih nacionalnih ciljeva koje ima ova zemlja sada. I da se vidi da, ako tekuća politička konstelacija naših odnosa sa svetom ne dozvoljava povezivanje naučno-istraživačke infrastrukture, svako odlaganje u tom povezivanju nosi potencijalno veće i veće gubitke za budućnost. I zbog toga se u ovoj politici mora, imajući na umu i sankcije, da se otvara uslovi za povezivanje naše infrastrukture sa međunarodnom, pogotovu sa onom koja ima pravi internacionalni karakter. Na primer, CERN.

Ideja „Sto vladinih stipendista" ušla je u zvaničnu proceduru.

Mislim da je apsolutno došlo vreme da se iz budžeta za ovu godinu, do jeseni, izabere tih sto mladih ljudi vladinih stipendista u zemlji i inostranstvu. Oni su vladin ljudski kapital. Ja u njima vidim sto ljudi koji će svake godine raditi za javni deo naučno-istraživačkog sistema Jugoslavije, koji ća se odlučiti da rade za potrebe ove zemlje i time se odredi potencijalnog odliva mozgova za narednih deset do petnaest godina. To rade sve valjane vlade i nema razloga da to ova ne uradi.

Postoje li finansijska sredstva za program povratka naučne emigracije?

Ta sredstva treba da budu, ako se radi o nivou Jugoslavije, u budžetu Jugoslavije za nauku i tehnologiju, ako je na nivou Srbije, u budžetu Srbije, Crne Gore i tako dalje. Za prve tri meseca budžet je obezbedio učešće naučno tehnološkog razvoja sa 0,4 % društvenog proizvoda. Sada je to prema najnovijem predlogu budžeta svedeno na 0,15 %, što me jako žalosti. Ta sredstva bi bila dovoljna da se prihvati populacija koja je emigrirala na način koji je ovde moguć. Ako hoće neko da se vrati, a nema gde da se vrati, savezna vlada kaže ili republička, evo tebi plata godinu dana, ti si na nivou docenta, dok ne nađeš sebi mesto kao docent ili kao istraživač tog nivoa u privredi, doći ćeš i primati platu na vladinoj blagajni. Prvo, takvi ljudi zaposlenje jako brzo nalaze. Drugo, ove godine i idućih nekoliko godina, a ja mislim i neko duže vreme, broj takvih ljudi će biti jako mali. Hoću da odgovorim na vaše pitanje koje nije postavljeno – povratak emigranata iz populacije naučno-istraživačkih radnika je jako mali svuda u svetu. Prave se specijalni programi na međunarodnom nivou, na primer, Evropska zajednica pravi specijalne programe koji su pre svega namenjeni Grčkoj, Španiji i Portugaliji – da se vrate njihovi naučni radnici. Koliko ja znam, iskustvo tih zemalja nalaže zaključak da se ljudi teško vraćaju.

Razlog za ostanak u emigraciji sigurno nije samo plata, već razvojni izazovi i okruženje.

Zemlje koje su uspele da vrate jedan deo svojih naučnika – Kina, Indija, Irak... vratile su ih zato što su im ponudile radikalno bolje uslove nego što su imali naučnici koji su ostali u zemlji. Mi to ne možemo ponuditi, čak naša kultura tu ponudu ne dozvoljava i mislim da je to potpuno u redu.

Bio sam prošle godine i pre dve godine na Tajvanu i video kako to oni rade. Vraćaju se da osnuju vrlo profitabilne i tehnološki intenzivne firme. Ima dosta naučnih radnika Tajvanaca koji su se vratili na Tajvan. U našu sredinu će se malo ko vratiti.

Poglavlje „Finansijski instrumenti" programa Savezne vlade predviđa oslobađanje od uvoznih dažbina za knjige, komunikacione i informacione uređaje, računare, faks, modeme i drugu opremu specifičnu za kućni rad naučno-istraživačkih radnika. Kada će to da profunkcioniše? I drugo, zar to što se oslobađaju od dažbine samo naučni radnici nije u koliziji sa onom narodnom da „bez alata nema zanata"? Zar ne treba dobra oprema i nekom ko tek treba da postane naučni radnik?

Taj instrument je do kraja moje čedo. Očekivao sam da će se usprotiviti ministar finansija, ministar za ekonomske odnose sa inostranstvom – ali nisu. Tako prolazio je. Moje je uverenje da onaj ko se bavi istraživanjem, bavi se istraživanjem i u Institutu i kod kuće. Ako oslobodimo, na primer, sikaru od uvoznih dažbina za uvoz boje, papira, štafelaja, isto tako treba da bude oslobođen i istraživač. Međutim, vi pitate – zašto ga oslobađamo tek kada postane istraživač. Zbog toga što ne znamo ko će postati istraživač, pa bi onda sve živo oslobadali i omogućili špekulaciju. Ako ste čitali dobro instrument, tamo piše: u slučaju da se oprema proda, moraju da se plate sve uvozne dažbine. Znači, instrument štiti državu od špekulacije.



Predvideli ste atašee za nauku i tehnologiju pri ambasadama Jugoslavije po svetu. Zar njihov posao ne može lakše da se obavi putem računarskih mreža, shodno individualnim odnosno, organizacionim potrebama?

To je nešto što sam je predlagao i u staroj Jugoslaviji kada sam bio član komisije za nauku Savezne skupštine. Videli ste, u prvo vreme računa se na pet atašea u zemljama odakle bi trebalo da crpimo najviše naučno-tehnoloških informacija i privrednih informacija koji su u vezi sa naukom i tehnologijom. Malo je zlurado rečeno, ali da je stara Jugoslavija umesto pet atašea druge vrste imala pet tehnoloških atašea, kao Finci ili Danci, stvari bi možda išle drugim tokom.

Vladin projekat predviđa baze podataka i informisanje uopšte. Dokle se stiglo na ovom planu i da li je Institut, kao jaka računarska kuća, uključen u to?

Savezno ministarstvo za nauku, tehnologiju i razvoj pravi programe i baze podataka, potpuno uređuje taj sistem. Novina je što u mnogim oblastima šire javnost ima pristup bazama. Mislim, pre svega, na baze o projektima koje finansira Savezna vlada. Svaki građanin može pristupiti bazi podataka i pitati ko je radio na projektu i šta je za te pare koje je dobio za projekat učinio za potrebe ove države. Mislim da je demokratski da postoji pozitivne povratne sprega, to je jedini način da imamo razumnu državu, da poreski obveznik ima uvid u to kako se poreski dinari troše. Naravno, time će jačati pritisak javnosti na naučno-istraživački sistem, ali mislim da je dobro. On će raditi pored kritike i vrlo pozitivan interes.

Šta sada u uslovima izolacije?

Prvo što bi trebalo u uslovima izolacije postići, i što je delimično iskazano preko pokušaja koje smo mi u vladi pravili, to je da se, kad nemamo sredstava da se bavimo istraživanjem na projektima, da istražimo kako ćemo to u budućnosti da radimo. Kod nas je politika naučno-istraživačkog razvoja pravljena suviše spontano – da damo svima po malo, da ne uznemiri nikoga, ili da uznemiravanje bude politički prihvatljivo. Seljak kada nema šta s jeseni i preko zime da radi, misli šta će, kako će na proleće da poseje najbolje. On to vreme ne prepušta prosto da teče beskonačno. Tako i preduzeće koje se nalazi u teškoćama pod sankcijama trebalo bi da smisli šta će da radi kada tražnja naglo poraste. Prvo treba sagledati budućnost. Kada su ljudi u velikom poslu, onda obično ne gledaju na dugi rok već ono što je neposredno pred njima. Dugi rok traži smirenje, traži poziv na mudrost.

Javni izvor finansiranja, znači republički, savezni budžeti, morali bi da finansiranjem dobrih projekata obezbede zaposlenost istraživačke populacije da se ne bi protraćile u zemlji, čak i ako ne odu ljudi u inostranstvo. Očigledno je da je i tamo gde je održano finansiranje smanjen nivo strogosti prema rezultatima tog finansiranja. Nije mali broj slučajeva gde se fondovi prazne, a ne

Nekada su odlazili samo mladi do 35, a sada odlaze i stari do 50

Ja se tome radujem jer ćemo mi usled tog pritiska morati ili da propadnemo ili da se podignemo na znatno viši nivo

pita se za rezultat. Ja mislim da je to najgora politika koja se može voditi jer ona rađa neodgovornost. Sankcije jesu – ali ako institut, fakultet, grupa ili pojedinac ugovori projekat sa republičkim, saveznim ministarstvom ili nekim drugim izvorom finansiranja, ima da da rezultate koji su predviđeni kao da ne postoje sankcije. Znači, naučno-istraživački sistem treba da sačuva svoju unutrašnju disciplinu i unutrašnju naučnu odgovornost.

Treća stvar koju treba raditi za vreme sankcija je obrazovanje ljudi. Plašim se da se to najmanje radi. Ako se plaćaju radnici jer su na prinudnom odmoru, bilo bi mudro obavezati ih da se, kad već primaju platu, usavršavaju svoj zanat, bilo koji da je, ili da uče nešto novo. Jer mi ne očekujemo da živimo pod sankcijama narednih sto godina. Plašim se da ćemo skidanje sankcija dočekati mnogo manje sposobni nego što bi inače mogli da bude. Mi u Pupinu to radimo. Ja sam insistirao, recimo, na učenju jezika. Svaki istraživač, svaki menadžer mora da nauči jezik. Formalno ćemo goniti ljude da uče. Da nam se ne desi kada izađemo iz sankcija da više ne umemo da pišemo pisma na engleskom jeziku. Ljudi će zaboraviti. Naši menadžeri dva sata godišnje poklanjaju učenju, a američki sto šezdeset sati. Na tome se zasnivalo i privredno čudo.

Verovatno najveća čovekova nesreća je kada ode u zatvor. Mnogim ljudima zatvor je donosio, ako se to uopšte može reći, izuzetne koristi. Imao je pred sobom ceo dan. Sabrao se, porazmislio, ako je imao sklonosti ka učenju, jako mnogo je naučio. Ove sankcije su nas stavile u situaciju koja je metaforično slična zatvoru. Šta onda da radimo? Jedna varijanta je da u zatvoru igramo karte je li tako? Druga varijanta je da učimo. Da ne budemo slepci!

Neću da vas pitam kako se razvijaju, već kako se u ovim uslovima održavaju računarstvo i komunikacije u Pupinu?

Ako nema tražnje, nema računarstva. Znači, program našeg računarstva je zaštićen delom time što postoji tražnja, a delom zbog toga što je smanjena ponuda budući da najveći broj stranih firmi nema sa Jugoslavijom više ništa.

Što se tiče telekomunikacionih programa, deo programa je takođe sačuvan zahvaljujući tražnji, na primer modemski program. Ovih dana smo sklopili jedan veliki ugovor iz te oblasti. Drugo, razlog je smanjena spoljna ponuda, a treći razlog – domaće snabdevanje u telekomunikacionom sistemu Srbije bilo je nikakvo. Telekomunikacioni sistem Srbije se snabdevao uglavnom iz Slovenije i nešto iz Hrvatske. Sada smo prosto primorani da razvijamo domaće kapacitete u ovoj oblasti. To, naravno, ima i pozitivan uticaj na Institut, koji je inače bio veoma malo angažovan, na primer, u razvoju domaćeg srpskog PTT sistema, što nije bilo dobro ni za nas ni za tog korisnika PTT. Kada budu skinute sankcije, mi ćemo imati veće probleme. Imaćemo šire prilaze svetu, ali ćemo imati veće probleme ove vrste. Strana ponuda će da poraste, što će biti pritisak na našu konkurentsku sposobnost. Ja se tome radujem jer ćemo mi usled tog pritiska morati ili da propadnemo ili da se podignemo na znatno viši nivo.

Vi mamv ulivate optimizam zato što kažete – kada budu uklonjene sankcije, ne kažete – ako budu uklonjene sankcije.

Skinuće se. Rat ima kraja. Ne može u Evropi da opstane zemlja predugo pod sankcijama. To može da se desi na rogu Afrike, može da se desi u Aziji, u Evropi to ne može. Mi smo, makar to nekad ne hteli, ipak deo i to sasvim primetan deo Evrope.

Sada, kada ste opet na čelu Instituta „Mihajlo Pupin" i kada se na određen način borite da plovite zajedno sa Institutom, ne ličite li malo sebi na Don Kihota?

Pa, mislim da bih i ja vama mogao to pitanje da uputim. Tu ste i niste otišli. Time hoću da kažem nešto drugo, metaforično, u smislu da smo svi mi Don Kihoti. Sa razvojnih mogućnosti koje su bile sjajne, pali smo na znatno niži nivo. Ako je išta u Jugoslaviji danas žalosno, to je gubitak razvojne pozicije koju je Jugoslavija imala u odnosu na istočni svet pre tri-četiri godine. Fascinantna je bila pozicija Jugoslavije. Mogla je da bude značajan most između Istoka i Zapada, između Severa i Juga, u tehnološkom razvojnom, čak i kulturnom pogledu. E, sad kada posle svega toga neko misli da i dalje nešto može da napravi, liči na Don Kihota. Međutim, tu smo gde smo.

KOLIKO BRZO RAČUNAJU

Na nedavno završenom Sajmu tehnike i građevinarstva autor ovog teksta je, kao i prošle godine, izvršio testiranje izloženih snažnih PC platformi. Testiranje je vršeno već ustaljenim programom, svojevremeno razvijenim kako bi se utvrdilo koliko brzo pojedine platforme rešavaju određen (veliki) sistem linearnih jednačina (takozvani LINPACK).

Bojan Tepavčević

Brzina sa kojom veliki sistem rešava linearne jednačine je po pravilu neuralgična tačka mnogih programa koji se bave, na primer, strukturalnom analizom (konačni elementi), a posebno kod nelinearnih problema, gde se sistem linearnih jednačina rešava u iteracijama. Kod većih problema, izvršavanje takvih programa, čak i na vrlo jakim platformama, može potrajati satima (pa čak i danima), tako da nimalo nije svejedno kakve brzine platforme pokažu na ovom testu, jer ubrzanje od nekoliko procenata znači minute ili čak sate u praksi.

RAZLIKE NISU MALE

Na brzinu izvršavanja ovog testa ne utiče brzina diska, video karte i drugih periferija, već pre svega brzina procesora i memorije, popunjenost memorijskih banki, stanja čekanja, način projektovanja matične ploče i, što će ovaj tekst i pokazati, vrsta i aktivnost eksternog keša. Pokazaće se da su na većini PC platformi moguće ogromne razlike u brzinama pri rešavanju velikih sistema linearnih jednačina, u zavisnosti od toga da li je eksterni keš uključen ili isključen.

Programi za testiranje razvijeni su u Fortranu, s tim da je za testove u realnom modu korišćen *Microsoft Fortran 5.00*, dok je za testove u *protected* modu korišćen *Microway NDP Fortran 386 v1.4*.

Suština testiranja se sastoji u rešavanju tri sistema linearnih jednačina, s tim da je matrica sistema simetrična, a elementi te matrice, koji su različiti od nule, grupisani oko glavne dijagonale. Rastojanje najudaljenijeg elementa matrice sistema (različitog od nule) od glavne dijagonale obično se naziva BAND. Prvi sistem sastoji se od 558 jednačina sa 558 nepoznatih i BAND-om 100. Za rešavanje ovog sistema dovoljno je 640KB RAM memorije. Drugi sistem ima isti broj jednačina, ali BAND 500, tako da je za izvršavanje programa koji taj sistem rešava potrebno 4MB RAM-a. Treći (i najveći) sistem ima 2081 jednačinu sa 2081 nepoznatom i band 500, a zahteva 8MB RAM-a.

Sam test je razložen u 5 faza. U prvoj, drugoj i trećoj fazi rešava se prvi sistem (558 jednačina, band 100). U prvoj fazi sistem rešava verzija programa koja radi u realnom modu i ne koristi koprocesor, a u drugoj verzija koja takođe radi u realnom modu, ali koristi koprocesor. Za treću fazu koristi se verzija programa koja radi u *protected* modu i koristi koprocesor. U četvrtoj i petoj fazi rešavaju se sistemi od 558 i 2081 jednačina sa BAND-om 500, a programi koji ovaj posao obavljaju takođe rade u *protected* modu i koriste koprocesor.

Ovaj tekst nema za cilj da pokaže koji je računar najbrži (dobijeni rezultati odnose se na dosta uzak deo primene – brz računar za ovu svrhu nekako se može pokazati kao spor za neku drugu). Ovo pre svega zato što mnogi izlagači nisu imali ni vremena ni mogućnosti da popune sve memorijske banke brzim memorijama, kao ni da pronađu minimum stanja čekanja sa kojima bi računar radio pouzdano, a sve to može bitno da utiče na konačni rezultat.

REZUTLATI TESTIRANJA...

priloženi su po fazama u 5 tabela. U prvoj i drugoj numeričkoj koloni tabela nalaze se vreme izvršavanja i indeks (u odnosu na 386/33) za računar sa isključenim eksternim kešom, dok se u trećoj i četvrtoj numeričkoj koloni nalaze vreme izvršavanja i indeks za isti računar, ali sa uključenim eksternim kešom. U petoj koloni nalazi se odnos vremena dobijenih sa uključenim i isključenim eksternim kešom, kako bi se direktno video procenat ubrzanja ili usporenja, u zavisnosti od toga da li je keš uključen ili isključen. Gotovo po pravilu eksterni keš će ubrzavati proračun u slučajevima kada se koristi manje RAM memorije (Faza 1, 2 i 3), zato što tada dobar deo memorije koju treba obrađivati može da se smesti u 256KB (što je, manje-više, standardni eksterni keš za 486), a usporavati (i to znatno) u fazama 4 i 5, gde je potrebno 4 odnosno 8 MB RAM-a. Ovaj fenomen se može dosta lako objasniti time da mnogi dizajneri eksternih keš-kontrolera stavljaju akcenat na što brži rad računara u realnom modu, pod DOS-om, pod pretpostavkom da će količina memorije koju je potrebno obraditi najčešće biti dovoljno mala da se u keš smesti. Međutim, kod programa ovakvog tipa, kada se rešavaju veliki sistemi jednačina, veoma brzo se manipuliše ogromnim blokovima memorije, tako da ona bukvalno samo "protiče" kroz keš, ne zadržavajući se preterano u njemu. U tom smislu, treba posebno pohvaliti matične ploče (tj. algoritme ugrađene u keš kontroler) kod kojih je odnos brzina pri uključenom i isključenom eksternom kešu što manji.

U sledećih pet tabela priloženi su rezultati (po goreopisanim fazama) za sve platforme testirane na ovogodišnjem Sajmu, kao i za neke platforme sa prošlogodišnjeg Sajma ili ranije, koje su se pokazale kao brze. U tabeli za fazu 5 nema onih platformi koje ili nisu imale dovoljno memorije za izvršavanje programa, ili jed

Tabela 1 – 558 jednačina, BAND 100, realni mod bez koprocesora

	Platforma	ext. off Time	ext. on Index	Time	on/ Index	off
1)	Šumadija 486DX2/66	12.77	3.78	12.41	3.88	0.972
2)	IMTEL 486DX2/66	12.65	3.80	12.44	3.88	0.981
3)	Cores 486DX2/66	12.71	3.79	12.47	3.87	0.983
4)	COMO 486DX2/66	13.12	3.66	12.52	3.85	0.955
5)	ADA Computers 486DX2/66	12.91	3.74	12.53	3.85	0.970
6)	ADA Computers 486/50	16.48	2.93	16.25	2.97	0.986
7)	MP biro 486/50	16.65	2.90	16.26	2.97	0.977
8)	Comtrad 486/50	16.62	2.90	16.29	2.96	0.980
9)	TECHNICOM 486/50	16.75	2.88	16.32	2.96	0.974
10)	BM 486/50	16.73	2.88	16.34	2.95	0.977
11)	Mikronova 486/50	16.67	2.89	16.36	2.95	0.982
12)	Micro Anima 486/50	16.89	2.86	16.40	2.94	0.971
13)	Ei Informatika 486/50	N/A	N/A	16.47	2.93	N/A
14)	GAMA m3 486/50	16.92	2.85	16.48	2.93	0.974
15)	Intel 486/50	16.75	2.88	N/A	N/A	N/A
16)	GAMA m2 486/33	24.60	1.96	24.27	1.99	0.987
17)	Jugodata 486/33	24.56	1.96	N/A	N/A	N/A
18)	Compaq SystemPro 486/33	N/A	N/A	24.30	1.98	N/A
19)	Everex step 486/33	N/A	N/A	24.60	1.96	N/A
20)	Intel 486/33	N/A	N/A	24.??	1.96	N/A
21)	NOVKABEL 486/33	25.12	1.92	24.80	1.94	0.987
22)	Mikronova DTK 486/33	N/A	N/A	24.83	1.94	N/A
23)	GAMA Notebook 486/33	N/A	N/A	24.96	1.93	N/A
24)	Cores 386/33 + Cyrix	N/A	N/A	48.23	1.00	N/A

Tabela 2 – 558 jednačina, BAND 100, realni mod sa koprocesorom

	Platforma	ext. off Time	ext. on Index	Time	on/ Index	off
1)	COMO 486DX2/66	5.16	4.29	4.33	5.11	0.839
2)	Šumadija 486DX2/66	4.72	4.69	4.38	5.06	0.927
3)	IMTEL 486DX2/66	4.65	4.76	4.41	5.02	0.948
4)	Cores 486DX2/66	4.83	4.58	4.43	4.99	0.917
5)	ADA Computers 486DX2/66	4.96	4.46	4.47	4.95	0.900
6)	ADA Computers 486/50	5.90	3.75	5.66	3.91	0.960
7)	Comtrad 486/50	6.00	3.68	5.68	3.90	0.945
8)	MP biro 486/50	6.02	3.67	5.68	3.90	0.942
9)	BM 486/50	6.08	3.64	5.69	3.89	0.937
10)	Ei Informatika 486/50	N/A	N/A	5.71	3.88	N/A
11)	TECHNICOM 486/50	6.08	3.64	5.71	3.88	0.940
12)	Micro Anima 486/50	6.15	3.58	5.71	3.88	0.928
13)	Mikronova 486/50	5.97	3.71	5.71	3.87	0.957
14)	GAMA m3 486/50	6.04	3.66	5.72	3.87	0.947
15)	Intel 486/50	5.86	3.76	N/A	N/A	N/A
16)	Compaq SystemPro 486/33	N/A	N/A	8.35	2.65	N/A
17)	GAMA m2 486/33	8.68	2.55	8.38	2.64	0.966
18)	Jugodata 486/33	8.41	2.63	N/A	N/A	N/A
19)	Everex step 486/33	N/A	N/A	8.46	2.62	N/A
20)	Mikronova DTK 486/33	N/A	N/A	8.51	2.60	N/A
21)	Intel 486/33	N/A	N/A	8.62	2.57	N/A
22)	NOVKABEL 486/33	9.08	2.44	8.70	2.54	0.958
23)	GAMA Notebook 486/33	N/A	N/A	8.98	2.46	N/A
24)	Cores 386/33 + Cyrix	N/A	N/A	22.13	1.00	N/A

Tabela 3 – 558 jednačina, BAND 100, protected mod

	Platforma	ext. off Time	ext. on Index	Time	on/ Index	off
1)	Sumadija 486DX2/66	2.31	6.46	2.14	6.98	0.925
2)	ADA Computers 486DX2/66	2.47	6.05	2.14	6.98	0.867
3)	Cores 486DX2/66	2.47	6.04	2.19	6.83	0.884
4)	IMTEL 486DX2/66	2.25	6.63	2.19	6.82	0.972
5)	COMO 486DX2/66	2.69	5.56	2.20	6.79	0.818
6)	Comtrad 486/50	2.92	5.11	2.80	5.34	0.958
7)	MP Biro 486/50	2.91	5.13	2.80	5.34	0.962
8)	GAMA m3 486/50	3.02	4.95	2.80	5.33	0.927
9)	Imtel 486/50	2.85	5.24	N/A	N/A	N/A
10)	Mikronova 486/50	2.91	5.14	2.86	5.23	0.983
11)	TECHNICOM 486/50	3.02	4.95	2.86	5.22	0.947
12)	ADA Computers 486/50	2.86	5.22	2.86	5.22	0.999
13)	BM 486/50	3.02	4.95	2.86	5.22	0.947
14)	Micro Anima 486/50	3.13	4.77	2.86	5.22	0.914
15)	Ei Informatika 486/50	N/A	N/A	2.86	5.22	N/A
16)	GAMA m2 486/33	4.23	3.53	4.17	3.58	0.986
17)	Imtel 486/33	N/A	N/A	4.17	3.58	N/A
18)	Compaq SystemPro 486/33	N/A	N/A	4.17	3.58	N/A
19)	Jugodata 486/33	4.18	3.57	N/A	N/A	N/A
20)	Everex step 486/33	N/A	N/A	4.28	3.49	N/A
21)	GAMA Notebook 486/33	N/A	N/A	4.34	3.44	N/A
22)	NOVKABEL 486/33	4.40	3.40	4.45	3.36	1.011
23)	Mikronova DTK 486/33	N/A	N/A	6.53	2.29	N/A
24)	Ceres 386/33+Cyrix	N/A	N/A	14.94	1.00	N/A

Tabela 4 – 558 jednačina, BAND 500, protected mod

	Platforma	ext. off Time	ext. on Index	Time	on/ Index	off
1)	Microway i860/33	N/A	N/A	33.71	11.39	N/A
2)	IMTEL 486DX2/66	57.23	6.71	78.98	4.86	1.380
3)	Sumadija 486DX2/66	58.22	6.58	79.37	4.84	1.363
4)	Cores 486DX2/66	61.85	6.20	82.38	4.66	1.330
5)	ADA Computers 486DX2/66	62.67	6.13	86.45	4.44	1.379
6)	COMO 486DX2/66	67.84	5.65	94.31	4.07	1.388
7)	ADA Computers 486/50	71.51	5.37	87.56	4.38	1.224
8)	Imtel 486/50	71.62	5.36	N/A	N/A	N/A
9)	Mikronova 486/50	72.34	5.31	88.49	4.34	1.223
10)	Comtrad 486/50	72.94	5.26	90.90	4.22	1.246
11)	MP Biro 486/50	72.98	5.26	90.78	4.23	1.244
12)	GAMA m3 486/50	75.31	5.10	101.61	3.78	1.349
13)	BM 486/50	75.46	5.09	101.78	3.77	1.349
14)	TECHNICOM 486/50	75.64	5.08	102.00	3.76	1.348
15)	Micro Anima 486/50	78.80	4.88	109.88	3.50	1.394
16)	Ei Informatika 486/50	N/A	N/A	83.16	4.62	N/A
17)	GAMA m2 486/33	106.44	3.61	105.35	3.64	0.990
18)	Jugodata 486/33	105.46	3.64	N/A	N/A	N/A
19)	Intel 486/33	N/A	N/A	105.56	3.64	N/A
20)	Compaq SystemPro 486/33	N/A	N/A	[illegible]	3.54	N/A
21)	GAMA Notebook 486/33	N/A	N/A	109.79	3.50	N/A
22)	Mikronova DTK 486/33	N/A	N/A	112.05	3.43	N/A
23)	NOVKABEL 486/33	112.10	3.42	145.61	2.64	1.299
24)	Everex step 486/33	N/A	N/A	124.51	3.08	N/A
25)	Cores 386/33+Cyrix	N/A	N/A	383.88	1.00	N/A

Tabela 5 – 2081 jednačina, BAND 500, protected mod

	Platforma	ext. off Time	ext. on Index	Time	on/ Index	off
1)	Microway i860/33	N/A	N/A	106.83	11.34	N/A
2)	IMTEL 486DX2/66	179.55	6.75	251.83	4.81	1.403
3)	Sumadija 486DX2/66	182.90	6.62	253.32	4.78	1.385
4)	Cores 486DX2/66	195.10	6.21	263.53	4.60	1.351
5)	ADA Computers 486DX2/66	197.18	6.14	276.99	4.37	1.405
6)	ADA Computers 486/50	224.04	5.41	276.60	4.38	1.235
7)	Imtel 486/50	224.43	5.40	N/A	N/A	N/A
8)	Mikronova 486/50	226.85	5.34	280.23	4.32	1.235
9)	Comtrad 486/50	228.55	5.30	287.70	4.21	1.259
10)	MP Biro 486/50	228.66	5.30	287.76	4.21	1.258
11)	BM 486/50	237.06	5.11	324.01	3.74	1.367
12)	Micro Anima 486/50	246.89	4.91	350.48	3.46	1.420
13)	GAMA m2 486/33	333.40	3.63	329.56	3.68	0.988
14)	Mikronova DTK 486/33	N/A	N/A	343.06	3.53	N/A
15)	GAMA Notebook 486/33	N/A	N/A	344.27	3.52	N/A
16)	NOVKABEL 486/33	351.58	3.44	457.04	2.65	1.300
17)	Ceres 386/33+Cyrix	N/A	N/A	1211.16	1.00	N/A

jednostavno nisu bila testirane za tu fazu (zbog dugog vremena izvršavanja testa).

EKSTERNI KEŠ: DA ILI NE?

Iz priloženih rezultata može se lako videti da na testovima koji rešavaju velike sisteme jednačina (tj. koriste velike količine RAM memorije) uključeni eksterni keš može da uspori proračun i za čitavih 40% dok kod manjih problema (koji zahtevaju manje memorije) doprinosi znatno manjem ubrzanju (tj. uglavnom između 5 i 10% a najviše 22% (što su i logični rezultati za efekat eksternog keša). Kada bi se ova disproporcija prenela u praksu, gde bi na primer bilo potrebno da se neki ogromni nelinearni problem proračunava rešavanjem sistema linearnih jednačina u iteracijama pod pretpostavkom da sa isključenim eksternim kešom proračun traje 10 sati (ili 10 dana što nije nemoguće) sa uključenim eksternim kešom isti problem bi se rešavao 14 sati (ili 14 dana). Pored toga odnos brzina pri uključenom/isključenom kešu nije konstantan i povećava se sa povećanjem RAM memorije koja se koristi (što se može videti u tabelama 4 i 5). Tako da bi, kada bi se koristilo 16, 32, 64 ili više megabajta (ukoliko postoje na ploči) razlike bile sve veće i veće.

U kontekstu ovog fenomena eksternog keša najinteresantnija je Gamina ploča 486/33 model2 (m2) koja je za razliku od ostalih na svim testovima pokazala za koji procenat bolje rezultate sa uključenim eksternim kešom (što bi na prvi pogled od svake ploče trebalo i očekivati) nego kada je radila sa isključenim eksternim kešom. To može da znači ili da je algoritam ugrađen u eksterni keš veoma dobar ili da keš uopšte ne funkcioniše.

Kao zaključak može se reći sledeće: probajte da isključite eksterni keš na svom računaru. Može lako da se dogodi da vam neki programi (koji rade u *protected* modu i koriste mnogo memorije) prorade znatno brže.

ŠTA IMA NOVO

Memorija

Nikad dosta RAM-a

Za pet godina činiće nam se da su današnje aplikacije koje nam zadaju sive muke svojim obimom u stvari imale sasvim skromne dimenzije. Novi tipovi digitalnih podataka (video, zvuk) i nove aplikacije (virtuelna stvarnost) će zahtevati ogromne količine sistemskog RAM-a samo za učitavanje, a još više za izvršavanje. "Toshiba", "Hitachi" i NEC su nedavno predstavili tehničku dokumentaciju o 256-megabitnom DRAM-u, čija se pojava očekuje krajem devedesetih.

Svaki čip može da sadrži 32 megabajta podataka – 64 puta više od današnjih čipova od 4 megabita. Novi čipovi će omogućiti veću gustinu i kapacitet. "Toshiba" koristi 0.4 submikronsku tehnologiju. Hitachi i NEC pripremaju 0.25-submikronski čip, što će omogućiti da se na silicijumsku pločicu smesti više ćelija za čuvanje podataka. Poređenja radi, Pentium koji ima 3.1 milion tranzistora koristi 0.8-submikronsku tehnologiju.

Akcija

COMTEC stipendije za talente

Redakcija naučnog i školskog programa TV NOVI SAD je pokretač, a novosadsko preduzeće za informatički inženjering i automatizaciju COMTEC d.o.o. sponzor akcije za stipendiranje studenata elektrotehničkog fakulteta, smer računarstvo i automatika. Raspisan je konkurs za dodelu dve stipendije za redovno školovanje studenata četvrte i pete godine ili apsolvenata. Pored ove dve predviđena je i jedna stipendija za boravak u inostranstvu. Naime, stipendista COMTEC-a će, uz plaćeno putovanje i sve ostale troškove, imati priliku da pohađa neki od kurseva koji se tokom leta održavaju na poznatim privatnim univerzitetima u Evropi. Izbor kursa, seminara ili letnjeg kampa koji bi bio obostrano interesantan zajednički će obaviti stipendista i COMTEC, dok će tehničku pomoć oko odlaska i prijave pružiti Redakcija naučnog i školskog programa TV NOVI SAD.

Za detaljnije informacije možete se obratiti COMTEC-u na telefon 021/27 181.

Digitalna fotografija

Skener za filmove: Elektronski „Nikon"

„Nikon", poznati proizvođač fotografske opreme, nedavno je na američkom tržištu plasirao svoj skener za filmove „Lajka" (35 mm) formata.

„Nikonov" skener radi kako sa negativima, tako i sa slajd (pozitiv) filmovima. Rezolucija je 2 700 dpi, a za beleženje informacija o svakom pikselu koristi se 24 bita. Prema navodima proizvođača, bez obzira da li se radi o crno-belom ili materijalu u boji, skeniranje se obavlja u jednom prolazu.

Interna verzija, odnosno ona namenjena za ugradnju u računar, odgovara dimenzijama poluvisokog nosača za diskove. Samostalna verzija je dimenzija 147×330×46 mm. Obe verzije koriste SCSI interfejs i sa verzijom za PC računare SCSI adapter i kabl služe u kompletu sa uređajem. U cenu paketa od 2 195 dolara za internu i 2 495 dolara za eksternu verziju pored hardvera je uračunat i softver za skeniranje.

MIŠEVI ZA NAJBOLJE PROGRAMERE JUGOSLAVIJE

Na Saveznom takmičenju iz programiranja koje je održano 29. maja 1993. godine u Beogradu, tri prva mesta zauzeli su Vladimir Petrovic, Božidar Radunović i Stanislav Šokorac – sva trojica učenici beogradske Matematičke gimnazije i sva trojica korisnici SEZAM-a – *vpetrovic, radunovic* i *ssokorac.*

Najbolji mladi programer Jugoslavije **Vladimir Petrović** osamnaestogodišnji je maturant Matematičke gimnazije u Beogradu. Programiranjem se bavi od šestog razreda osnovne škole, kada je dobio svoj prvi računar – *Commodore 64*. Uz njega je naučio bejzik, a kasnije je prešao na asembler. U srednjoj školi – koju je, kako sam kaže, završio sa lakoćom, jer nije bio opterećen predmetima koji ga ne zanimaju – prvo je savladao paskal, pa zatim C++, a u četvrtom razredu prolog. Sada ima računar 286 klase. Vladimir priča da je školsko takmičenje bilo teže proći nego gradsko: „Morali su proći prvo našu školu, jer inače niko drugi ne bi imao šansu na gradskom". Međutim, priznaje da su zadaci koje je za republičko takmičenje pripremio Jožef Kralica bili veoma teški – iz oblasti teorije grafova, za koju se kandidati nisu pripremali. Ipak, rešenje do kojeg je došao rekurzijom donelo je Vladimiru prvo mesto. Sa takvim iskustvom i uz malo ponavljanja teorije grafova Vladimir je pobedio i na Saveznom takmičenju. Kaže da mu je ta pobeda donela i bolji uspeh u školi – četvorke iz matematike se pretvorile u petice.

Osim za vlastite potrebe, pisao je i nekoliko programa za Matematički Institut. A njegov drug Božidar dodaje da su programi veoma zanimljivi da će biti objavljeni u inostranstvu. Kada ne sedi za računarom, Vladimir trenira aikido.

Drugo mesto je zauzeo njegov drug iz odeljenja, osamnaestogodišnji **Božidar Radunović**. U drugom razredu Matematičke gimnazije Božidar je učestvovao na školskom i gradskom takmičenju, u trećem je stigao do republičkog, a sada je na Saveznom njegovo rešenje bilo veoma blizu najboljeg. I Božidar nam je ispričao da su zadaci na republičkom takmičenju bili preteški – on je uspeo sa još 16 takmičara da se plasira dalje zahvaljujući samo dobrim idejama koje je žiri uvažio. Tek kasnije je savladao teoriju grafova. Posle takvog iskustva Božidar može da kaže da je na Saveznom takmičenju „jedan zadatak bio veoma lak, a drugi lakši nego onaj na republičkom."

Svoj prvi računar, spectrum, Božidar je dobio u drugom razredu osnovne škole. Uskoro je naučio bejzik i nakon prepisivanja programa za igrice počeo i sam da programira, najpre da bi sebi olakšao rešavanje domaćih iz matematike. Posle računara XT u srednjoj školi je prešao na 386 i savladao paskal i C++. Softver koji je u saradnji sa još nekoliko programera napisao za evidenciju studenata na PMF-u ove godine je prodat i Filozofskom fakultetu. U slobodno vreme sluša muziku i, kao i Vladimir, trenira aikido. Školski drugari će zajedno i na studije računarske tehnike na Elektrotehničkom fakultetu.

Sedamnaestogodišnji **Stanislav Šokorac**, trećeplasirani na Saveznom takmičenju, takođe je učenik Matematičke gimnazije u Beogradu, ali završava tek drugi razred. Ispričao nam je da su zadaci na školskom i gradskom takmičenju bili suviše jednostavni i nezanimljivi, dok ga je uspeh na republičkom iznenadio. Na Saveznom, što se zadataka tiče, za najbolje nije bilo iznenađenja.

Stanislav je od drugog razreda pa sve do kraja osnovne škole imao *Spectrum*-a, da bi ga u gimnaziji zamenio sa 286-icom. I on je počeo sa igricama, pa sa programiranjem u bejziku. U srednjoj školi je naučio paskal i otada ga programiranje sve više zanima – Sezamovcima su poznate avanture o SEZAM-u SUPI I III, čiji je autor *ssokorac*. Tokom leta ga očekuje pisanje softvera za *Olivetti* sa još četvoricom drugara iz Gimnazije. U slobodno vreme Stanislav voli da čita naučnu fantastiku, sluša muziku, igra tenis, go i pomalo šah.

Nagrada za prvoplasiranog je fax-modem i originalni softver *Turbo C++ 3.0*, a za drugog i trećeg po jedan miš, dok su sva trojica dobila diskete, nekoliko knjiga i, nezvanično, kao *post festum* česitku naše redakcije, šestomesečnu pretplatu na SEZAM. Osim toga, IMTEL je školi koju pohađaju pobednici poklonio računar 386. Vladimir, Božidar i Stanislav kažu da će konačno, uz desetak 286-tica, i jedan 386 biti dostupan đacima-običnim smrtnicima.

Najlepše od svega je što postoje dobri izgledi da naši najbolji mladi programeri učestvuju na Programerskoj olimpijadi koja se u oktobru ove godine održava u Argentini. Argentinska vlada im je uputila zvanični poziv, a kažu da se već javilo više sponzora zainteresovanih da finansiraju njihov odlazak na Olimpijadu. Oni sami su optimisti – rekli su da će nam nešto više o sebi ispričati posle Argentine.

Renka Jovanović



Najbolji u zemlji: Vladimir Petrović, Božidar Radunović i Stanislav Šokorac

SOFTVER

Visual Basic 3.0

U novoj verziji podrška za baze podataka

Sredinom maja „Microsoft" je na tržište izbacio *Visual Basic 3.0*. Nova verzija, prema rečima autora softvera, još više je prilagođena zahtevima koje su postavili najčešći korisnici *Visual Basic*-a – programeri u velikim korporacijama, čiji je zadatak razvijanje specifičnih aplikacija koje rade pod *Windows*-ima. Drugim rečima, *Visual Basic 3.0* omogućava lak pristup bazama podataka, zahvaljujući ugrađenom kodu za upravljanje bazama preuzetom iz programskog paketa *Microsoft Access 1.1*. Takođe, podržano je i povezivanje aplikacija napisanih u *Visual Basic*-u sa komercijalnim programskim paketima preko *OLE 2.0* standarda. U novoj verziji, sada već možemo reći popularnog programskog jezika, među su našli i novi alati kontrole, povećana je lakoća korišćenja i proširene mogućnost prilagođavanja kontrola konkretnim potrebama aplikacije koja se razvija.

Visual Basic 3.0 i sistem za upravljanje bazama podataka *Microsoft Access* sada na izvestan način predstavljaju familiju alata za rad sa bazama podataka. Sa jedne strane, *Visual Basic 3.0* omogućava razvoj specifičnih aplikacija koje u punoj meri koriste prednosti *Windows* radnog okruženja, dok sa druge strane *Microsoft Access* pruža mogućnost interaktivnog rada nad bazama podataka koje su formirane ili se održavaju pomoću programa napisanih u *Visual Basic*-u.

Ključna inovacija u *Visual Basic*-u *3.0* je, svakako, *Microsoft Access 1.1 Database Engine*, koji omogućava direktni pristup bazama podataka formiranim u najpopularnijim formatima kao što su *dBASE*, *Paradox*, *Btrieve* i *MS Access*. Tu su i ODBC (*Open Database Connectivity*) drajveri za *Microsoft SQL Server*, *Sybase SQL Server* i *ORACLE*. Prema kazivanju „Microsoft"-ovih projektanata, obezbeđena je robusna podrška višekorisničkom radu, UNDO funkcija nad bazama, širok spektar tipova podataka sa kojima se može raditi (obuhvaćeni su, pored standardnih, i zvuk, video, slike i OLE objekti).

Microsoft Access 1.1 Database Engine omogućava rad sa bazama podataka čija je veličina preko jednog gigabajta, a pristup funkcijama je moguć kako kroz programski, tako i kroz vizuelne objekte. I upravo tu se krije možda najveća snaga ovog paketa. *Data Access Objects* je skup objekata koji omogućava lako programiranje sistema za upravljanje bazama podataka: definisanje i kompletno manipulisanje baze. Programerima *Data Access Objects* omogućavaju rad sa upitima, pojma, indeksima ili tabelama.

Data-aware Controls je deo *Microsoft Access Database Engine*-a koji je „zadužen" za direktni kontakt korisnika sa bazom podataka. Znači, radi se o kontrolama koje povezuju vizuelne objekte (okvire sa tekstom, slike, polja za unos i izmenu podataka...) sa bazama. Naravno, oba dela *Database Engine*-a se lako mogu međusobno povezati.

Kada se govori o radu sa bazama podataka, jedna od nezaobilaznih stvari su izveštaji. Njihovo lako kreiranje u paketu *Visual Basic 3.0* omogućava *Cristal Reports 2.0 for Visual Basic*, koji upravo koristi *Microsoft Access 1.1 Database Engine*. Sada programeri mogu redizajnirati izveštaje korišćenjem popularne WYSIWYG tehnike. *Cristal Reports for Visual Basic* podržava formule, kalkulacije u dva prolaza, grupisanje podataka bez ograničenja, podešavanje izveštaja neposredno pred štampanje, *Print Preview*, bitmapirane slike, velike tekstove i još mnogo toga. Za programere to znači da pomoću *Cristal Reports*-a mogu postavljati izbor formula, naslove, sortirati podatke, određivati koje grupe podataka će biti prikazane u izveštaju, kao i određivati odredište izveštaja – prozor na ekranu, štampač ili datoteka na disku.

Od novih alata izdvajamo *Database Manager*, program napisan u samom *Visual Basic*-u *3.0* (izvorni kod se može naći na „CompuServe"-u), pomoću kojeg se može manipulisati već kreiranim bazama podataka. Tu je i *Setup Wizard*, alat koji u velikoj meri automatizuje proces izrade instalacionih procedura.

Visual Basic 3.0 dobio je i nove kontrole, pomoću kojih se mogu kreirati hijerarhijski „list box-ovi" slični onima u *Windows File Manager*-u. I druge kontrole, naročito za obradu unosa i izmene podataka, su dorađene u odnosu na prethodnu verziju *Visual Basic*-a. Kada već govorimo o kreiranju korisničkog interfejsa, napomenimo da se sada *pop-up* meni mogu kreirati u samo jednoj liniji izvornog koda. U ovom segmentu usavršena je i obrada informacija koje prosleđuje miš. Naime, i desno dugme na mišu sada se mnogo više koristi nego ranije.

Microsoft Visual Basic 3.0 isporučuje se u dve verzije – *Standard Edition* i *Professional Edition*. Osnovna razlika među njima je što *Standard Edition* u sebi ne sadrži ODBC, deo objekata koji omogućavaju programski pristup bazama podataka, *Cristal Reports* i deo kontrola koje su i ranije bile privilegija *Professional Edition* verzije. Hardverski zahtevi za rad sa *Visual Basic*-om 3.0 su instalirani *Windows 3.0* ili viši, hard-disk, EGA, VGA ili Hercules, 2 MB RAM-a za razvojno okruženje (preporučuje se dvostruko veća memorija), odnosno 1 MB za računare na kojima će se koristiti aplikacije napisane u ovom programskom jeziku. M.B.

Nastavak sa strane 18

800x600 sa 65000 boja) mach32 takođe odbija da radi pod *Windows*-ima u određenim rezolucijama – mašina u kojoj je instalirana ova kartica se naprosto zaglavljuje.

NEKA DRUGA PRIČA

Analize rada grafičkih kartica u *Windows* okruženju bi se mogle vršiti nadugačko i naširoko, pri čemu bi trebalo uložiti puno truda da se uoči formula po kojoj se odnose režim rada kartice, brzina procesora, arhitektura matične ploče. *Windows* postavlja izuzetno velike i složene zahteve pred resurse čitave mašine, a naročito pred grafičke kartice. Ti zahtevi su često vrlo protivrečni – osobine koje su neophodne za brz rad sa velikom paletom boja nemaju nikakve veze sa funkcijama za tehničko crtanje – otuda nije lako napraviti univerzalnu grafičku karticu koja će podjednako dobro moći da odgovori na zahteve raznih kategorija softvera, odnosno da zadovolji razne kategorije korisnika. Ovo pravilo posebno važi za grafičke akceleratore koji svoju brzinu zahvaljuju upravo posebnoj optimizaciji za određenu vrstu grafičkih poslova. Mi smo posmatrali ponašanje grafičkih kartica u svim režimima za koje su bili obezbeđeni *Windows* drajveri, ali smo za prikaz rezultata i poređenje izabrali tri tipična režima – 640×480 sa 16,7 miliona boja, 800×600 u 65000 boja i 1024×768 u 256 boja – koje podržavaju kartice na svim testiranim platformama.

Rezultati koje smo dobili potvrđuju priličnu osetljivost modernih grafičkih adaptera za pojedine vrste poslova, što znači da treba dobro razmisliti ne samo koji grafički adapter staviti u svoju mašinu nego i u kom grafičkom režimu pokrećete koju aplikaciju. Sve VESA kartice su postigle prilično rezultat u radu sa velikim brojem boja, da bi se favorit izdvojio tek u režimima sa velikim brojem tačaka. Iz *Wintach* dijagrama je jasno da se od kartica sa Cirrus čipom mogu očekivati ujednačeni rezultati na koje neće mnogo uticati ostatak platforme, pod uslovom da je 486 i pa te da je Diamond Stealth Pro apsolutni favorit. Zanimljivo je da je u režimu 1024x768 kartica Stealth 24 za ISA brža od Cirrus kartica za VESA sabirnicu, i da sa ISA mach32 više nego dobro nosi sa VESA karticama u režimima sa velikim brojem boja. Da bi bilo jasnije o kakvim poboljšanjima je reč, dajemo i rezultate za klasičan grafički adapter sa karticom Trident 8900 u mašini 486/50 MHz. Poboljšanja kod VESA kartica se kreću od **dve** do **preko pet** puta.

Situacija izgleda nešto drugačije kada se pogledaju rezultati za tipične kategorije aplikacija. Kartica mach32 koja briljira u testovima slikanja i CAD crtanja, u rezoluciji 640x480 u 16,7 miliona boja jedva je zbog sporosti privela kraju test iz obrade teksta. Dok smo posmatrali sliku na ekranu kako se gubi u netrivijalnijim grafičkim operacijama činilo nam se da nas je neko vratio u predistoriju računarstva. Oni koji koriste *paint* i CAD programe mogu slobodno da se odluče za ovu karticu i da ne razmišljaju mnogo o VESA sabirnici. Ako prelazite na novu arhitekturu, Stealth Pro će vam obezbediti nešto manje pogodnosti i skromniju paletu ali uravnoteženije osobine kroz sve režime i aplikacije, podjednaku brzinu u radu sa velikim paletama i znatno brži rad sa velikim brojem tačaka na ekranu, posebno u programima CAD i *paint* tipa. Odaberite pravi režim i – spremite se da poletite.

DOSTA JE BILO BROJKI

Nema je navavno kao svakom korisniku bio najzanimljiviji test sa programima sa kojima svakodnevno radimo i za koje procenjujemo da bi se prijatnije osećali na snažnijem hardveru. Ulazak u omiljeni *XyWrite* je trenutan, ulazak u *Windows* i rad pod njim brži, prijatniji. *QuarkXpress* reaguje brzo kao da je u pitanju neki mali DOS program a ne složena *Windows* aplikacija, uz *CorelDraw* smo se manje doradivali čekajući da izvede najsloženije grafičke operacije (i da nacrta čuvenu Kolibu pod snegom [20-30 sekundi prema 90 na mašini sa ET4000 čipom], pomeranje i povećavanje crteža u *AutoCad*-u je bilo trenutno, manipulacija slikama i složene transformacije u *PhotoShop*-u pristojno brze, promena taskova u *Windows*-u ili korišćenje OLE tehnika glatko i potpuno bezbolno. (Jedino za *Venturu* još nije izmišljen dovoljno brz hardver. Možda kad dođe Pentium.) Ovu vrstu testova, međutim, svaki kupac može i treba da uradi za sebe. Mi bismo, kao korisnici ovih mašina, nekom manje nekom više, bili uglavnom prilično – zadovoljni. Bilo je veliko zadovoljstvo gledati novu tehnologiju na delu.

Iako broj računara sa kojima smo radili nije dovoljan da bismo sa sigurnošću mogli da utvrdimo, prvi utisak je da nova arhitektura PC mašina sa lokalnom sabirnicom u VESA standardu ispunjava svoja spektakularna obećanja. Ali stvar, naravno, nije samo u brzoj magistrali nego i u novoj arhitekturi grafičkih adaptera. Da li ćete ostati na ISA ili preći na VESA arhitekturu, videćete uskoro i sami. To vas, ako se sa VESA standardom ne dogodi ništa neprijatno, niko neće ni pitati. Naprosto, u jednom trenutku više nećete imati izbora.

NIJE NAM TEŠKO

ComTrad Group je organizacija osnovana u Venecueli 1984. godine. Aktivnost kompanije je proizvodnja i prodaja informatičke opreme i usluga, pod imenom „ComTrad".

ComTrad YU

ComTrad YU trpi teret sankcija kao i druge firme u Jugoslaviji. Veze sa ComTrad Group zamrznule su na neodređeno vreme, povezanost sa centralom ostala je u ovom trenutku samo po imenu firme i programu računarske opreme koju ComTrad YU nudi. Kako je uvođenje sankcija prestalo da bude novost i pre nego što su službeno stupile na snagu, jugoslovenski ComTrad reagovao je na vreme uvozom količina opreme dovoljnim za poslovanje tokom dve poslovne godine. Kompanija se oslanja na sopstvene resurse, kako kaže **Veselin Jevrosimović**, jedan od vlasnika ComTrada. Grupa ne sme biti povezana sa jugoslovenskom operacijom, a firma u Beogradu mora biti spremna za nastavak poslovanja punom snagom istog momenta kada se poboljša politička situacija. Konkurentnost ComTrad opreme ogleda se na više načina: visokim kvalitetom, cenama koje stalno padaju i praćenjem najnovijih trendova.

Kao deo grupe, ComTrad YU nudi PC računare 386SX, 386 i 486, „Epson" i „Hewlett Packard" štampače, „Novell" mrežni softver, UPS uređaje, mrežnu opremu kao i sve vrste prateće opreme. U skladu sa svetskim trendovima, „ComTrad" je još prošle godine iz ponude izbacio mašine sa 286 procesorom, kao i mašine sa monohromatskim (MGP, „Hercules") grafičkim adapterima i monitorima. U ponudi se nalazi sa i cela gama „Canon" crno-belih i kolor uređaja za foto-kopiranje, bubble-jet štampača, fax aparata, kao i „CASIO" kalkulatora i registar kasa.

Kao svetski specijalist za LAN i WAN računarske mreže, ComTrad izvodi sve vrste umrežavanja („Novell" *NetWare*, „Microsoft" *Windows for Workgroups*, PC računari, UNIX mašine, mini računari, *mainframes*, *packet networks*, *internetworking*).

U toku proizvodnog procesa računari su izloženi rigoroznom testiranju komponenti da bi na kraju prošli kroz sistemski burn-in test.



ComTrad na Majskim danima tehnike u Beogradu Technotronica 93

ComTrad YU na sajmu računara u Beogradu

Sajam računarske tehnike u Beogradu od 11. do 15. maja ove godine bio je prilika da se dobije uvid u stanje računarske industrije (manje) i prodaje računara (nešto više). Štand ComTrad YU uneo je malo optimizma u inače ne-baš svetlu sliku. Firma je očito nastupila punom snagom, profesionalno i estetski vrhunski izveden štand odudarao je od postavki ostalih izlagača. Vesele šare koje znamo sa propagandnih materijala ComTrad YU davale su prepoznatljiv imidž prostoru. Šemung je doprinosila glasna muzika, spotovi firme na velikim video monitorima i naravno, recimo i to, privlačne domaćice štande.

Novitet kod računara koje je prikazao ComTrad YU bila su nova kućišta. 386SX računari prikazani su u elegantnim *slimline* kućištima, dok je rešenje *tower* i *mini tower* kućišta takođe izmenjeno, sada su to vrlo moderne, spreda zaobljene kutije sa bočnim otvaranjem. Na monitorima se ne štedi, proizvođač je „Daewoo" sa kojim grupa ostvaruje dugogodišnju saradnju. Asortiman štampača sadržavao je „Epson" matrične štampače i „Hewlett Packard" ink-jet i laserske štampače.

Mreža sa serverom i petnaest stanica radila je pod *NetWare* 3.11 uz *Windows for Workgroups*. Multimedia je bila prisutna kroz dva projekta: PC SoftSound audio stanica i stanica za baze podataka sa video zapisom. PC SoftSound je sistem za koji je softver razvijen u saradnji sa „Pingvin" radiom. Zvučni signal snima se i reprodukuje pomoću najnovije Sound Blaster 16ASP zvučne kartice sa profesionalnim kvalitetom. Namenjen je produkciji i emitovanju radijskog programa, prvenstveno komercijalnog.

Istorija grupe

ComTrad je počeo sa radom u Karakasu u Venecueli u prvom kvartalu 1984. godine. Ako pogledamo to razdoblje, setićemo se da je to tek četvrta godina nakon pojave računara IBM PC i da procvat proizvođača klonova i kompatibilnih računara tek dolazi. Taj val zapljusnuo je i Venecuelu, koja je pod direktnim uticajem američkog tržišta. Nakon uspostave lanca prodajnih i proizvodnih objekata u Venecueli osniva se centar za proizvodnju i distribuciju u SAD, da bi se optimizirali troškovi i logistika korporacije. Od tog trenutka organizacija se internacionalizuje i usvaja najsavremenije proizvodne, prodajne i marketinške metode za istovremeni nastup na tržištima više zemalja.

Kao sledeći korak uspostavljaju se tesni kontakti sa centrima za proizvodnju komponenti u Aziji, a posebno na Tajvanu. Prisustvo na sajmovima širom sveta i bolje upoznavanje globalnih uslova tržišta, uz optimizaciju troškova unutar firme i prema proizvođačima, uverava ComTrad da je sposoban da se probije i na dodatnim tržištima. Donosi se odluka o formiranju firmi u SAD, Španiji, Nemačkoj, Holandiji. Sredinom 1989. godine grupa odlučuje da se usvoji samo jedno trgovačko ime – ComTrad – i jedinstveni logotip za sve svoje proizvode, publikacije, kataloge, marketing, sa efektom kreiranja uniformnog imidža firme svuda u svetu.

Grupa zapošljava više od 700 ljudi u oko 40 firmi u više od 10 zemalja sveta. Grupa je prisutna u Venecueli, SAD, Tajvanu, Nemačkoj, Holandiji, Španiji, Jugoslaviji, Češkoj, Mađarskoj, Bugarskoj, Rusiji, Sloveniji i drugim zemljama. Centar za zapadnu hemisferu nalazi se u Majamiju, SAD. Evropu pokriva centrala u Diseldorfu, dok je ured u Beogradu, pored jugoslovenskog tržišta, zadužen za Istočnu Evropu. Proizvodnja je organizovana u Majamiju i u Češkoj. Osnivanje proizvodnog i distributivnog centra u Češkoj pravi je mali podvig, jer je u jako kratkom vremenu prenesena kompletna tehnologija ranije instalirana u Majamiju, uz maksimalno usavršen sistem kontrole kvaliteta.

DRUGI O NJIMA

CHS – novi član ComTrad Group

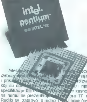
ComTrad, distributer PC računara sa sedištem u Majamiju, sa uredima u južnoj i istočnoj Evropi, upravo je kupio CHS, distributera Hewlett Packard proizvoda ( ).

Ovogodišnja promet čini ComTrad jednim od 20 najvećih distributera u Evropi ( ).

Carstvo ComTrada obuhvata Next DS, Microsoftovog distributera u Portugalu, distributivne firme u Nemačkoj, Holandiji, Češkoj, Mađarskoj, Bugarskoj, Rumuniji i bivšoj Jugoslaviji. Članice firme uglavnom prodaju ComTrad PC klon koji se za Evropu proizvodi u Češkoj.

„Osnovani smo pre sedam, a profit prikazujemo zadnjih šest godina. Zadovoljni smo načinom na koji se firma razvija", kaže Antonio Boccalandro, predsednik kompanije. Boccalandro takođe kaže da su najveće operacije u Češkoj, Nemačkoj i Portugalu; sam portugalski ured ostvaruje promet od preko 12 000 000 USD godišnje.

(Prevod iz „PC Europa" newsletter iz januara 1993)



ZVEZDA JE ROĐENA

Novi grafički operativni sistemi nalik na OS/2 i *Windows NT* zahtevaju sve više i više procesorske snage, koju 80486 sve teže obezbeđuje. *Intel* u pravom trenutku izlazi na tržište sa procesorom nove generacije, dva i više puta bržim od prethodnika. Umesto već prilično neorginalnog 80586, odabrano je ime *Pentium*, nova zvezda koju će 1993. godina urezati na (kompjutersko) nebo!

Intel [illegible] procesora izbacio [illegible] najavljen i prika[illegible] proizvođači koji su do[illegible] i njegove specifikacije bi[illegible] zasnovane na njemu ne prezent[illegible] pre 17. maja. Radilo se zapravo o jednoj od najbolje čuvanih tajni kompjuterske industrije: strogi embargo na specifikacije učinio je da javnost sa velikim nestrpljenjem sačeka 17. maj kada su se sva nagađanja o *Pentiumu* pretvorila u činjenice ili zablude. Znalo se jedino da će *Intel* za početak isporučivati dve verzije *Pentiuma* (60 i 66 MHz) zasnovane na 0.8 mikrometarskoj 5-voltnoj BiCMOS tehnologiji sa preko 3 miliona tranzistora. *Intel*-ov produkcioni ciklus tokom poslednjih 15 godina (8086 se pojavio na tržištu 1978. godine) krunisan ovim mikroprocesorom je po svemu izuzetan – u proseku su izbacivali po jednu generaciju mikroprocesora svaka 44 meseca, dok se broj tranzistora ugrađenih u njihove čipove penjao eksponencijalnom linijom. 8086 je bio zasnovan na 3-mikrometarskoj tehnologiji a *Pentium* na 0.8-mikrometarskoj, uz tendenciju daljeg poboljšanja tj. prelaska na 0.65 mikrometarsku 3.3-voltnu tehnologiju koja bi obezbedila rad na 100 i više megaherca, već je današnji *Pentium* pretendovao da bude 150 puta brži od najbržeg 8086!

Prvi nezavisni testovi pokazuju da sva ova očekivanja nisu bila preterana – prema *Dhrystone 1.1* testu, rejting 66 MHz *Pentiuma* je 112 MIPS-a (miliona instrukcija u sekundi) a 60 MHz 100 MIPS-a (poređenja radi, 486DX2/66 na istom testu postiže oko 54 MIPS), dok *SPECint92* skor iznosi 64.6, *SPECfp92* skor 59.7 i *iCOMP index* 567. Probijena je, dakle, magična barijera od 100 MIPS-a koja je tako dugo „mučila" porodicu 80x86 i prikazivala prednosti RISC-a nad raznoraznim CISC arhitekturama.

NA 273 NOGE

Pentium je poveći čip površine 14 cm^2, ograđen u 273-pinsko PGA (*pin grid array*) kućište. Njegovih 3.100.000 tranzistora, radeći na 66 MHz troše oko 13 W. Proizveden je pomoću 0.8 mikrometarskog BiCMOS procesa i obezbeđuje internu kontrolu parnosti i detektovanje grešaka pri keširanju instrukcija i podataka, asocijativnu keš memoriju i keš TLB (*translation lookaside buffers*) uz obavezni ROM sa mikrokodom. U njega su uz sve to ugrađena i kola koja obezbeđuju povezivanje i paralelni rad više jednakih *Pentium* čipova. Set instrukcija je praktično identičan naredbama 80486 uz sitne izmene koje na primer omogućavaju sistemskom programeru da implementira MESI (*modified exclusive shared, invalid*) multiprocesorski keš koherencijski protokol. Novitet su i neke 8-bajtne „uporedi i zameni" instrukcije, kao i nova naredba koja obezbeđuje identifikaciju procesora (šteta što tako nešto nije ugrađeno i u ranije članove ove porodice jer će identifikacija i dalje morati da se obavlja pomoću relativno složenih programa koje smo objavili u „Računarima"). *Pentium* ima i jedan kontrolni registar više od prethodnika – CR4 smo, kada smo proučavali arhitekturu 80486, već pomenuli kao „buduću ekspanziju". Uveden je najzad novi *System Management* mod rada – nešto slično *Intel* je već isprobao kod 386SL.

Dejan Ristanović

Rad procesora se zasniva na petostepenom izvršenju instrukcija. Imena faza se nisu promenila od procesora 80486: *prefetch, decode, address generate, execute* i *writeback*. Čim se tokom izvršavanja neke od instrukcija kompletira jedna faza, automatski se prelazi na sledeću, čime je napravljen prostor da sledeća instrukcija pređe u prevaziđenu fazu. Razlika između 80486 i *Pentiuma* je u tome što novi čip ima **dve** paralelne linije izvršavanja instrukcija. *Pentiumova* jedinica za rad sa celim brojevima u isto vreme uzima i dekodira dve instrukcije, proverava mogu li se one izvršiti paralelno i onda ih, ako je moguće, izvršava. *Intel* je za instrukcije koje se mogu izvršiti paralelno usvojio termin *simple instructions* koji ćemo mi, po analogiji sa međusobno prostim brojevima, prevesti kao „međusobno proste instrukcije".

MEĐUSOBNO PROSTE INSTRUKCIJE

Uprošćeno rečeno, međusobno proste su one instrukcije kod kojih izvršenje druge ne zavisi od rezultata izvršavanja prve. *Intel* je toj definiciji dodao zahtev da za izvršavanje ne bude potreban mikroprogram i da se svaka instrukcija može obaviti u jednom ciklusu. Izuzetak su instrukcije za prenos podataka u memoriju i iz memorije koje zahtevaju tri ciklusa, ali zahvaljujući specijalnom hardveru mogu da budu međusobno proste sa nekim drugim instrukcijama. Iz ovakve definicije je jasno da su, na primer, ADD EAX, EBX i POP EDX međusobno proste instrukcije jer prva operiše sa registrima EAX i EBX a druga sa EDX. Instrukcije ADD EAX,EBX i ADD EAX, EDX očito **nisu** međusobno proste jer se rezultat prvog sabiranja koristi u drugom. Međutim, postoje i mnoge suptilnije situacije u kojima nije lako reći da li su instrukcije međusobno proste ili nisu – SHL EAX, 1 i INC EBX deluju na razne registre, ali je stanje flegova po izvršenju ove sekvence kombinovano – prva je formirala CF a druga OF, SF, ZF, AF i PF, da su instrukcije izvršene obrnutim redosledom, SHL bi formirala sve flegove. Iz dostupne dokumentacije nije baš jasno da li *Pentium* ovakve instrukcije tretira kao međusobno proste ili ne (mada je u jednoj od notica tehničke dokumentacije navedeno da je U linija bazično „odgovorna" za flegove) – verujemo da ih tretira, jer kad ne bi bilo tako, onda i ne bi ostalo mnogo međusobno prostih instrukcija!

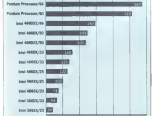
U svakom slučaju, kada „primeti" da su dve instrukcije međusobno proste, *Pentium* ih upućuje u U i V linije, od kojih svaka sadrži nezavisnu jedinicu za rad sa celim brojevima (ALU) i one se paralelno izvrše. Ukoliko se jedna instrukcija zbog kontrole prekida izvršava duže nego što je planirano, mikroprocesor će automatski usporiti izvršavanje druge da se ne bi izgubila sinhronizacija. Ako par instrukcija nije međusobno prost, kontrolna jedinica će poslati prvu od njih u U liniju i pokušati da upari sledeću sa narednom instrukcijom koja čeka na red. U i V linije nisu identične – prva obuhvata i takozvani *barrel shifter* koji obezbeđuje manipulaciju sa bitovima, što znači da samo ova linija može da izvrši **sve** *Pentium* instrukcije. V linija je nešto jednostavnija i prilagođena instrukcijama koje bi uopšte mogle biti međusobno proste sa nekom drugom naredbom. U linija je privilegovana i u tom smislu da se samo u njoj može izvršiti prva faza instrukcije koja radi sa racionalnim brojevima. Tek potom se ona prosleđuje „koprocesoru" (obe linije ipak mogu da pripreme FPU operand).

Ugradnja druge aritmetičko-logičke jedinice za rad sa celim brojevima (ALU) praktično duplira propusnu moć procesora, ali istovremeno zahteva dvostruko brže „dobavljanje" operanada – šta vredi što se operacija može izvršiti u paraleli ako njeni argumenti nisu spremni? *Intel* je zato udvostručio interni keš na 16 kilobajta i podelio ga na dva odvojena dvo-asocijativne zone, istovremeno je proširena interna i eksterna magistrala za podatke na 64 bita. Zanimljiv problem predstavlja i pribavljanje podataka iz keš memorije. Većini instrukcija potreban je, naime, bar po jedan operand, a većina tih operanada se nalazi u kešu. Šta se događa ako svaki od ALU a zatraži po jedan podatak, i to u isto vreme? Verovali ili ne, dobiće ga bez problema – TLB (*translation lookaside buffer*) i asocijativne oznake su smeštene u dvopristupnu (*dual port*) memoriju pa im obe ALU jedinice istovremeno pristupaju. *Intel*, na žalost, ovde nije bio baš dosledan tj. nije smestio kompletan keš u dvopristupnu memoriju, umesto toga formirao je „interliv" keš memoriju od 8 zona čije granice moraju biti na adresama deljivim sa četiri. Sve dok obema instrukcijama ne zatreba baš **isti** podatak (što je malo verovatno, ako su instrukcije uzajamno proste pa se izvršavaju zajedno) obe će ga istovremeno dobiti. Dobitak u performansama potiče i od toga što je keš memorija zasnovana na *write back* arhitekturi, za razliku od konzervativne *write-through* tehnike primenjene na 80486.

PROGNOZA GRANANJA

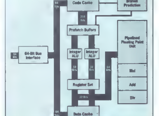
Što se instrukcija tiče, *Intel* je tazi prilikom dodao i prognozu grananja (*branch prediction*). *Pentium* zapravo ima dva 32-bajtna *prefetch* bafera, u svakom trenutku jedan od njih sekvencijalno obrađuje sekvencu instrukcija sve dok ne naiđe na neku naredbu uslovnog grananja. U tom trenutku se, na osnovu sadržaja BTB-a

Arhitektura procesora „Pentium"



Superskalarna arhitektura sa redukovanim setom instrukcija, dinamičkim predviđenjem grananja, proširenom matematičkom jedinicom, dva keša, *write* baferima i 64-bitnom magistralom za podatke (izvor: Intel)

iCOMP indeks



Cela porodica na brzinskom testu. Mašinska brzina Intelovih procesora u ICOMP (Intel Comparative Microprocessor Performance) jedinicama (izvor: Intel)

(*branch target buffer*) u koji su upisane informacije o prethodnim skokovima, vrši predikcija – da li će instrukcija uslovnog skoka stvarno proizvesti skok ili ne. Kada se, recimo, radi o klasičnoj FOR NEXT petlji, instrukcija skoka se „uglavnom" izvršava, a samo jednom od možda više stotina ili hiljada prolazaka kroz nju obezbedi izlaz iz petlje. Mikroprocesor će, dakle, sa velikom verovatnoćom predvideti da će do skoka doći, pa će u drugom *prefetch* baferu krenuti priprema instrukcija na mestu do koga se skokom dolazi. Ako je bila u pitanju petlja sa izlazom na vrhu, pa je mikroprocesor „uočio" da do skoka obično ne dolazi, nastaviće se sekvencijalno pripremanje instrukcija u glavnom *prefetch* baferu. Na taj način se postiže da posle svake ispravne procene mehanizam paralelnog izvršavanja uopšte ne bude poremećen skokom. Pogrešne procene (ili čak skok na neočekivanu adresu usled samomodifikujućeg koda) naravno znače da su za izvršenje pripremane „nepotrebne" instrukcije, pa će taj posao (najmanje tri ciklusa) biti „knjižen" kao neminovni gubitak.

Posebna pažnja posvećena je samomodifikujućim programima – već duže vreme teoretičari govore da je samomodifikujući kod sa raznih aspekata veoma loša stvar koju treba po svaku cenu izbegavati. Pa ipak, mnogi programeri iz raznih razloga posežu upravo za takvim tehnikama, što je „smrt" za integrisanu keš memoriju i preklapanje faze pripreme i faze izvršenja instrukcija. Konstruktori *Pentiuma* tvrde da su preduzeli posebne mere da se samomodifikujući kod detektuje i korektno tretira, naravno uz neizbežan (mali) gubitak na performansama. Razumevanje i kopiranje ove tehnologije (mora se implementirati identičan algoritam da bi samomodifikujući programi jednako radili) će za konstruktore eventualnih *Pentium* klonova biti veoma ozbiljan, ako ne i nerešiv problem.

RAD SA RACIONALNIM BROJEVIMA

Veliki broj ugrađenih tranzistora dolazi do punog izražaja kada se radi o operacijama sa racionalnim brojevima. *Pentiumov SPECfp92* skor od golovo 57 je bar 3,5 puta veći od rezultata koje je postizao 80486, što se pre svega pripisuje postojanju potpuno odvojenih jedinica za sabiranje, množenje i deljenje. Svako sabiranje i množenje će, bez obzira na zahtevanu preciznost, biti završeno za svega tri ciklusa, dok je deljenje nešto sporije – za svaka dva bita količnika troši se po jedan ciklus. Poređenja radi, kod 80486 je „obična" FADD instrukcija (sabiranje) „trošila" 10 ciklusa, a FMUL (množenje) 12 do 15 ciklusa.

Preklapanje instrukcija za rad sa racionalnim brojevima sa „običnom" celobrojnom aritmetikom je izuzetno dobro rešeno – izvršavanje neke od Fxxx instrukcija obavlja se u osam faza, pri čemu su prve četiri faze potpuno identične sa istoimenim fazama pripreme i izvršenja svih ostalih instrukcija. *Intel* na žalost nije obezbedio paralelno izvršavanje dve aritmetičke instrukcije, pre svega zato što nema dovoljno linija kojima bi pribavljao operande. Mali ali važan izuzetak je instrukcija FXCH koja, kao što će se setiti čitaoci naše serije tekstova o programiranju 80486, razmenjuje sadržaj bilo kojeg FPU registra sa vrhom aritmetičkog steka. S obzirom da ova instrukcija efektivno ništa ne računa, može je izvršiti *Pentiumova* V linija, i to u paraleli sa (gotovo) svakom instrukcijom integrisanog „koprocesora". Zahvaljujući tome bilo koja instrukcija za rad sa racionalnim brojevima može se izvršiti a njen rezultat prenet u „FPU akumulator" odakle će ga „preuzeti" sledeća instrukcija, sve to u minimalnom mogućem vremenu.

ŠTOS JE U KOMPAJLERU

Sve u svemu, glavna razlika između 80486 i *Pentiuma* je u tome što se prvi sastojao od jedne linije za izvršavanje instrukcija i keša koji se koristio kako za naredbe tako i za podatke. *Pentium* je ubrzao rad odvojivši keš za instrukcije i keš za podatke i dodavši drugi *pipeline* kojim se instrukcije izvršavaju. Sve magistrale su dvostruko šire da bi se obezbedila veća propusna moć. Jedinica za rad sa racionalnim brojevima je znatno bolja, brža i glomaznija. Sve u svemu, nastala je jedna arhitektura koja sa naučne tačke gledišta ne predstavlja naročitu novost (većina postojećih RISC procesora je već superskalarne sa paralelnom obradom celih i racionalnih brojeva, dok je dvostruka linija obrade instrukcija već viđena na *Sun Microsystems SuperSparc* i *Motorola 88110* stanicama), ali je upravo revolucionarna kada se radi o dizajnu jednog čipa iz tako kompleksne porodice kao što je 80x86. Ugraditi tolike elemente najmodernijeg dizajna a ipak sačuvati punu kompatibilnost sa 15 godina starim osmobitnim „pretkom" to može samo *Intel*!

Činjenica da je kompatibilnost očuvana ne znači da se pažljivim korišćenjem raznih „RISC tehnika" ne može postići i mnogo više od nje. Pogledajmo, recimo, program od dve linije

```
e=a+b+c
x=x+y+z
```

Uz pretpostavku da su svi podaci prethodno smešteni u registre i „zaboravljajući" na flegove i prekoračenja, ovaj bi se program mogao prevesti otprilike ovako (malo je neobično da se ESI i EDI koriste kao registri opšte namene, ali ovo je samo jednostavan primer):

```
ADD EAX,EBX
ADD EAX,ECX
ADD EDX,ESI
ADD EDX,EDI
```

Prilikom izvršavanja ovog segmenta *Pentium* neće moći da preklopi prve dve instrukcije, preklopiće drugu i treću i onda samostalno izvršiti četvrtu. Sve u svemu, tri ciklusa. Međutim, ako instrukcije (i dalje zanemarujući njihovo dejstvo na flegove) grupišemo nešto drugačije

```
ADD EAX,EBX
ADD EDX,ESI
ADD EAX,ECX
ADD EDX,EDI
```

dobijemo situaciju da se prva i druga instrukcija mogu izvršiti u paraleli, a isto tako treća i četvrta. Sve u svemu, uštеđen je jedan ciklus, odnosno dobijen program koji se izvršava za trećinu brže! Pokazuje se, na žalost, da „živ čovek" ne može da piše ovako optimizovane programe – nema

problema kada se, kao ovde, radi o nekoliko instrukcija, ali u programima od samo nekoliko desetina hiljada mašinskih naredbi ove „egzibicije" bi učinile rezultat toliko nečitljivim i teškim za održavanja da bi se malo ko odlučio za takvu avanturu. Na sreću, tu su optimizacioni C kompajleri – čovek piše program onako kako mu odgovara, a prevodilac ga transformiše u oblik koji najviše odgovara mikroprocesoru. *Intel* tvrdi (a mnogi nezavisni analitičari potvrđuju) da će se prevođenjem postojećih DOS i *Windows* aplikacija uz pomoć nekog od kompajlera prilagođenih *Pentiumu* dobiti i do 30% brži programi, bez ikakvih izmena u izvornom kodu! S obzirom da je gotovo sav komercijalno raspoloživ softver za PC računare pisan u C-u (ili u poslednje vreme C++-u), čak i mala poboljšanja kompajlera mogla bi da daju izvanredne rezultate. Ne može se, na žalost, očekivati da će se te nove verzije programa pojaviti sledećih meseci – vodećim proizvođačima kompajlera svakako će biti potrebno neko vreme da se prilagode (premda treba primetiti da je *Borland* još u februaru predstavio novu verziju C++ a specijalno prilagođenu *Pentiumu* i OS/2, a *Microsoft* je obećao *Visual C++* za *Windows NT*), a zatim će autorima softvera biti potrebno vreme da se prilagode tim kompajlerima. U međuvremenu bi broj računara u kojima se krije *Pentium* trebalo da postane dovoljno veliki da se takve unapređenja softvera uopšte isplate.

„Pentium" u brojevima

	i486DX2/66	i486DX/50	Intel Prototype 66MHz Pentium	ALR Evolution V 60MHz Pentium	ALR ProVEISA 60MHz Pentium (Funnel Design)
Processor Harmonic	14 929.6	13 481.8	30 509.6	26,020.5	26 268.8
16-Bit Prot. Mode Small Mix	116,786.0	92,924.0	239 646.0	209 260.0	215 192.0
16-Bit Real Mode Small Mix	125 782.0	101 593.0	241 889.0	214 127.0	217 700.0
16-Bit Prot. Mode Standard Mix	472.4	428.9	990.0	840.0	840.0
16-Bit Real Mode Standard Mix	499.5	465.1	1,009.9	852.1	865.6
Math Coprocessor	19 353.6	14 233.6	38 502.4	32 768.0	33 177.6

Rezultati brzinskih testova procesora u stvarnim uslovima (izvor: Computer Shopper)

NA BRZINSKOM TESTU

Pentium nije zamišljen samo kao „sledeći" mikroprocesor iz familije 80x86 – on je zadivljujuće kombinacije nasleđenih slabosti i (ne)očekivanih unapređenja. Do sada smo ga uglavnom procenjivali po osnovu unapređenja i kompatibilnosti, ostaje da vidimo kako čitava kombinacija deluje u praksi. Pri proceni performansi *Pentiuma* najbolje je kao reper koristiti postojeći 80486 DX2-66 – oba mikroprocesora pripadaju istoj porodici i interno rade na istoj frekvenciji, što znači da je svaka razlika u performansama rezultat unapređenog dizajna. Prvi testovi niskog nivoa (za aplikativne testove je, zbog nedostatka pravih *Pentium* ploča, još rano) koje su sproveli nezavisni autori pokazuju ubrzanje od 1.6-1.8 puta u odnosu na *Compaq DeskPro 66M*, 66-megahercni „super PC" zasnovan na 486DX2. Kada se radi o operacijama sa racionalnim brojevima, performanse su čak i bolje – zahvaljujući boljoj integraciji komponenti i unapređenoj arhitekturi o kojoj je bilo reči, *Pentium* je 1.9 puta brži od 486DX2.

Učinjen je i pokušaj testiranja na razvojnim pločama čija arhitektura znatno bolje odgovara *Pentiumu*, tj. ne „koči" njegov pristup memoriji i periferiji kao što ga koče 486 ploče na koje je *Pentium* „nakrpljen". Performanse su sa popele na 2 – 2.2 puta veće vrednosti od 486DX2, što je svakako izuzetan rezultat koji nagoveštava da bi se aplikativni brzinski testovi mogli izvršiti za 1.7-1.9 puta kraće vreme od istih tih testova na najbržim 80486 pločama. Sve se, dakle, vrti oko magične formule kojom se svaka *Intel*-ova generacija ponosila: „mikroprocesor (bar) dvostruko brži od prethodnika"!

KAKO DO PENTIUMA?

Proizvođači prvih *Pentium* sistema su, kao što se i očekivalo, uglavnom giganti: *ALR, AST Research, Compaq Computer, Hewlett Packard, IBM, NCR, Unisys* i *Zenith Data Systems*. Sve te firme su nestrpljivo čekale 17. maj da bi u ranim jutarnjim časovima pokušale da budu prve sa *Pentium* pločama – prema informacijama koje smo uspeli da sakupimo do zaključivanja ovog broja „Računara", najbrži su bili *Advanced Logic Research* (ALR), *Dell Computer Corporation* i *Compaq Computer*. ALR nudi najjeftiniju mašinu – zajedno sa diskom od 170 megabajta njihov računar košta oko 3 000 dolara. Cena *Dell*-ovog *4560/XE file server*-a, zasnovanog na 60 MHz *Pentiumu*, iznosiće oko 5 500 dolara (plus cena monitora), za osnovni sistem sa diskom od 320 megabajta. Objavljena je i cena unapređenje postojećih *Dell PowerLine 486/SE* servera – oko 2500 dolara. *Compaq* je najskuplji, ali i najbolji – zahvaljujući *TriFlex/PC* arhitekturi koja komunicira preko tri odvojeno optimizovane magistrale: 128-bitnoj memorijskoj i 64-bitnoj procesorskoj magistrali i ulazu/izlazu preko 32-bitne EISA (*extended industry standard architecture*) sabirnice, računari *Compaq Deskpro 5/66M* i *Compaq Deskpro 5/60M* biće nešto najbrže što je PC svet video. Po ceni koja nije mala ali nije ni astronomska, za one kojima treba takva moć – najjeftiniji model košta 5000 dolara, ali će mnogi poželeti i *QVision 1280* grafički kontroler (1 280×1 024 u 256 boja), 20-inčni monitor *QVision 200*, više memorije, veći disk... cena brzo dostiže 10 000 dolara. *Pentium* se verovatno neće pojaviti i na sceni portabl računara: firma *Lightning Computers* je 20. maja počela da isporučuje „super portabl" zvani *ThunderBox II* – trenutno u njemu radi 80486 na 66 MHz, ali je tu i podnožje za *Pentium*, pa čim uspeju da ga nabave...

Ta nabavka, na žalost, neće biti baš jednostavna. Trenutna proizvodnja *Pentium* čipova je veoma mala, a obimnije isporuke se ne planiraju pre oktobra. *Intel* planira da posle toga postepeno povećava proizvodnju, da bi do kraja 1993. isporučio 400 000 komada, a već sledeće godine čak 2 ili više miliona.

Biće, osim toga, i problema pri dizajniranju i serijskoj proizvodnji matičnih ploča u koje će se ugrađivati *Pentium*. Problemi su se, zapravo, pojavili već i kod ploča u kojima 80486 radi na 50 megaherca (486SX2 koji radi na 66 megaherca je manji problem jer je 66 MHz interna brzina u procesoru – čitava ploča i dalje radi na 33 MHz). Na brzinama od 60 i više megaherca nema nikakva margine za komponente sumnjivog porekla ili nemaran dizajn. Ako se frekvencije popnu na 100 i više megaherca, što je, rekli bi se, neminovno, cena *Pentium* kompjutera neće pasti onako brzo kao cena 80386 i 80486 računara. Pored raznih „preslušavanja" TTL signala i varijacija u naponu, najveći problem predstavljaće grejanje, bolje reći pregrevanje čipa. Jedan ventilator više u kompjuteru neće biti dovoljan, a od proizvođača će se očekivati da pažljivije proračunaju kretanje vazduha kroz kućište – ugradnja *Pentium* ploče u neko staro kućište lako može dovesti do pregrevanja mikroprocesora!

Zbog svih tih problema, mnogi proizvođači oprema još ne nude kompletne sisteme zasnovane na *Pentiumu* nego samo kartice za proširenje njihovih postojećih računara. Cena prvih kompletnih Pentium sistema zavisiće i od toga da li se radi o „prešminkanom" 80486 računaru ili o pravoj arhitekturi prilagođenoj novoj generaciji. SMP (*The Symmetrical Multiprocessing*) arhitektura koju nude NCR i AST bi trebalo da omogući paralelni rad više *Pentiuma*, ali bi cena takvih mašina lako mogla da pređe 30 000 dolara.

Sve se, dakle, vrti oko cene, u kojoj popriličnu učestvuje i cena samog čipa – *Pentium* će, ako naručite količinu od preko 1000 komada, koštati 965 odnosno 878 dolara za verzije na 66 i 60 MHz. Nije poznato kada će *Pentium* moći da se nabavi „na komad" i po kojoj ceni, ali se očekuje da te cena bude znatno viša. Ako vam sve te cifre izgledaju astronomski visoke, setite se da prvi primerci starog Z-80 nisu koštali mnogo manje od toga, a danas se Z-80 prodaje za dolar-dva!

Već je poznata i jedna varijacija *Pentiuma* čije je kodirano ime P24T – radi se zapravo o nekoj vrsti *Pentium Overdrive*-a, odnosno o unapređenju postojećih 80486 sistema. Neke postojeće 80486 ploče, na primer AST-ov *Premmia 4/66d*, imaju posebno podnožje sa dodatnim redom pinova za P24T. Kada se početkom sledeće godine pojavi na tržištu, P24T će se umetati u 238-pinsko podnožje, zamenjujući postojeći procesor. Na žalost, većina ovako „friziranih" računara neće ponuditi brzinu dostojnu „pravog" *Pentiuma*, pošto će P24T komunicirati sa sekundarnom keš memorijom preko 32-bitne magistrale na samo 33 MHz, dok *Pentium* sistemi kompletnu komunikaciju obavljaju 64-bitnim linijama i to punom brzinom. Da bi se ovo usporenje izbeglo, skuplja P24T proširenja će biti instalirana na posebnu karticu koja će obuhvatati i eksterni keš.

KONKUREKCIJA NE SPAVA

Ima li alternative *Pentiumu*? Mnogi će, poučeni ranijim iskustvima, reći da nema gore konkurencije od konkurencije sličnih proizvoda. Ipak, *Intel* ne bi trebalo previše da se pribojava AMD-a i *Cynx*-a, koji su nekada tako uspešno klonirali 80386 – njima još nije uspelo da proizvedu ni pravog 80486 klona! AMD-ov „486" projekat je „pao" prošlog decembra kada je, kao što su „Računari" pisali, federalni sud presudio u korist *Intel*-a, potvrdivši da ugovor o saradnji iz 1976. godine ne daje pravo AMD-u da koristi *Intel*-ove mikroprograme. Tako je 80486 klon, koji je AMD najavljivao za početak ove godine, neslavno nestao sa tržišta – AMD će morati da napiše svoj mikrokod, što će trajati više meseci, verovatno duže od najavljenog roka početkom jula.

Cynx je tokom 1992. godine predstavio tri „486" procesora: Cx486SLC, Cx486DLC i Cx486S2/50. Sva ova imena, međutim, ne znače da je ijedan od ovih čipova dostojan 80486 klon – set instrukcija je manje više isti, ali nema integrisana jedinica za rad sa racionalnim brojevima (FPU, „koprocesor"), interni keš je sveden na jedan ili dva kilobajta (pravi 80486 ima integrisani keš od 8 kilobajta). *Cynx*-ov „pravi" 80486 klon

zvani M7 najavljen je za drugu polovinu 1993 ali je pitanje kada će se pojaviti i da li će opravdati očekivanje.

Ako je kloniranje 80486 bilo (i ostalo) priličan problem, kloniranje *Pentiuma* bi se lako moglo pretvoriti u noćnu moru. Problem se zapravo može „napasti" na dva načina – detaljno proučiti i kopirati *Pentiumovu* arhitekturu ili dizajnirati „od nule" sopstveni mikroprocesor koji će imitirati njegovo ponašanje, naravno uz iste ili bolje performanse. Posle presude iz decembra 1992. AMD i *Cyrix* su se definitivno opredelili za drugu varijantu. AMD, zapravo, razvija novu 0.35 mikrometarsku tehnologiju koja bi trebalo da mu obezbedi oko 10 miliona tranzistora u istom čipu, što bi rezultiralo procesorom u klasi *Pentiuma* ili čak nečim boljim. Takav razvoj je, međutim, nesumnjivo spor, pa se prvi rezultati očekuju tek 1995. godine i kasnije, uz troškove koji prevazilaze milijardu dolara. Finansijska strana bi delimično trebalo da se reši AMD-ovim sporazumima sa *Hewlett Packard*-om, koji radi na novoj generaciji PA-RISC procesora. *Cyrix* je, čini se, definitivno odustao od *Pentium* klona i za ovu ili sledeću godinu najavljuje „konkurentni" 100-150 MIPS čip koji će biti bolje optimizovan, tj. neće zahtevati rekompilaciju postojećeg softvera u cilju poboljšanja performansi. Sve to su, po mnogim analitičarima, „kule u vazduhu".

Konkurencija izgleda neće doći ni od tradicionalnog takmaca *Motorole* – iako je serija 68000 još od 1970-tih konkurent *Intelovoj* 80x86 familiji, izgleda da je *Motorola* zakasnila čitavu godinu dana za *Pentiumom*: 68060 je planiran za 1994. godinu, a njegove obećane performanse (50 MHz i 77 MIPS-a) ne izgledaju naročito privlačno.

Reklo bi se, dakle, da *Intel* više nema konkurencije i da je razvojno i tehnološki toliko superioran u odnosu na sve druge proizvođače mikroprocesora da će sam ući u sledeći vek. Međutim, ne mora tako biti – *Intel* je zbilja pobedio sve koji su mu konkurisali na njegovom polju, ali se pojavljuju novi čipovi koji nisu opterećeni kompatibilnošću sa čitavom jednom porodicom mikroprocesora, koja i dalje ime svoje 8-bitne predstavnike i čija arhitektura i dalje mora da se zasniva na već arhaičnim pretpostavkama. Proizvođači novih (pretežno RISC) mikroprocesora mogu da puste na volju svojim idejama i tako proizvedu čip koji će možda imati mnogo manje tranzistora i mnogo jednostavniju arhitekturu, za koji u startu neće biti tržišta, ali će na duži rok ponuditi više od *Pentiuma*. Najozbiljniji pretendent na to je projekat *PowerPC*, nova linija RISC čipova koju proizvode *Motorola* i IBM u saradnji sa firmom *Apple*. Ovi mikroprocesori počivaju na IBM-ovom *RISC System/6000* procesoru i pretenduju da na njima bude zasnovana sledeća generacija IBM-ovih i *Apple*-ovih personalnih računara. Značajnu reč (i to u oba smera!) daće i proizvođači operativnih sistema – *Microsoft* nagoveštava da će *Windows NT* raditi na raznim procesorima, dok se *NeXT*, *Solaris* i *Sun* približavaju 80x86 familiji. Još jedan partnerski poduhvat IBM-a i *Apple*-a, *Taligent*, za 1995. godinu najavljuje operativni sistem koji će se izvršavati na raznim platformama; jedino još nije najavljen OS/2 za ne-*Intel*-ove sisteme.

Mnogi smatraju da će se RISC/CISC „rat", bar kada se radi o personalnim računarima, obaviti i okončati na *Windows NT*-u. NT je, reklo bi se, „prirodan" operativni sistem koji će se izvršavati na *Pentium* sistemima, ali, po prvi put, ne samo na njima. Korisnici će najzad biti u prilici da nabave PC na kojem se izvršava NT i koji uopšte nije zasnovan na *Intel*-ovom mikroprocesoru – možda će *Windows NT* jednako dobro ili bolje raditi na, recimo, DEC-ovoj *Alpha* seriji ili na *Silicon Graphics/Mips R4000*. *Windows NT* obezbeđuje simetrično multiprocesiranje, što znači da će *Pentium* morati da se suoči sa odnosom cena/performanse koji nude, recimo, sistemi sa 2-3 paralelno vezana 80486 procesora. Pa ipak, *Pentium* ima tri izuzetno jaka aduta. Pre svega, veliko je pitanje koliko će brzo i koliko pouzdano NT raditi na novim platformama – *Microsoft*-u se, realno govoreći, već više puta desilo da obeća znatno više nego što docnije ostvari, pa je kupovina ovakvog sistema, nasuprot proverenoj 80x86 porodici, svojevrsno kockanje bez šansi za naročito veliki dobitak. Drugo, postojeći DOS i *Windows* programi izvršavaće se samo na mikroprocesorima iz porodice 80x86 (premda se mogu zamisliti i 8086 emulatori, ali tek uz značajan gubitak na performansama). Najzad, iako proizvođači RISC procesora ne moraju da brinu o kompatibilnosti sa starim prethodnim generacijama, pa su prema tome slobodni da primene sva dostignuća nauke, ni jedan od njih ne može računati na *Intel*-ov nivo proizvodnje koji bi garantovao dovoljno niske cene čipova.

Najzad, ni *Intel* neće spavati dok konkurenci je radi – već je u toku projekat koji treba da dovede do *Pentiumovog* naslednika čije je radno ime P6. Neki nezavisni izvori kažu da će se P6 pojaviti na tržištu za 18-20 meseci, da će sadržati preko 10 miliona tranzistora i obezbediti najmanje 200 MIPS-a, tj. biti bar dvostruko brži od *Pentiuma*. Detalje o eventualnoj 128-bitnoj ili čak 256-bitnoj arhitekturi i novom setu instrukcija, *Intel* je za sada nazvao „spekulacijama".

Sve u svemu, reklo bi se da je, probivši granicu od 100 MIPS-a, *Intel* ozbiljno uzdrmao kompletnu RISC produkciju i to tukući je njenim oružjem – superskalarnom arhitekturom i paralelizmom!

KO SU PRVE MUŠTERIJE?

Najveće tržište za *Pentium* u početku će, kao što to obično biva, biti mrežni serveri i grafičke radne stanice, a zatim sledi tradicionalni „napad" na tržište miniračunara i čak *mainframe* sistema, koje *Pentium* po mnogim karakteristikama prevazilazi. Kada se radi o „običnom korisniku" koji obrađuje tekst, sortira podatke, računa, crta i tome slično, ponavlja se priča koju smo već više puta čuli: kada je izašao AT, pričalo se da je „običnom čoveku" dovoljan XT; kada su se pojavile prve 386 mašine, govorilo se da je AT dovoljan za većinu primena; pojava 80486 je rezultirala tvrdnjama da je 80386 na 33 megaherca sasvim dovoljan čak i za korisnika većih zahteva. Kako cene padaju, tako se pojavljuju i potrebe, a kao što „prilika stvara ugodu", tako i „ugoda stvori priliku" – „korisnik većih zahteva" primeti da mu nova generacija obezbeđuje obavljanje istog posla za dvostruko manje vremena i počinje da razmišlja o unapređenju. Takvom razmišljanju pogoduju programi koji se pojavljuju svakog dana i koji, istina, rade i na 80386/80486, ali prilično sporo, dok su na *Pentiumu* „munjeviti". Najzad, korisniku ne preostaje drugo nego da se maši za novčanik, a ubrzo se pojavi i sledeća generacija mikroprocesora koja opet izgleda tako nedostižno. To je, izgleda, neizbežni krug u koji smo upali kada smo za 50 funti kupili *Sinclair ZX 80*!

U ovom trenutku se, posebno uzevši u obzir naš standard, teško može savetovati kupovina računara zasnovanog na *Pentiumu* – malo ih je na tržištu, pouzdan rad na 66 MHz je pod znakom pitanja, nisu se formirale standardne matične ploče, otvoreno je pitanje magistrale, nisu isključeni procesorski „bagovi" i, što je najvažnije, preskupi su! Pa ipak, sledeća godina će na naše stolove doneti *Pentium* kompjutere, možda čak i brže nego što bi neko danas prorekao. U međuvremenu, uživaćemo u tome što će 80486 računari, zbog pojave sledeće generacije mikroprocesora, znatno pojeftiniti!

ŠTA IMA NOVO

DOMAĆA SCENA

Mediji: meki zvuk Pingvina

Znaju li za ovo i vaši slušaoci

U vašem srcu Pingvin, u srcu Pingvina kompjuter, u srcu kompjutera – radio!

Zahvaljujući ideji popularnog beogradskog radija „Pingvin", softverskoj mašti tima mladih Pingvinovih programera i hardverskoj podršci kompanije COMTRAD, radio budućnosti je sada samo stvar jednostavne odluke: hoćete li ili nećete da pođete Pingvinovim stopama i program vaše radio stanice automatizujete uz pomoć hardversko-softverskog paketa nazvanog *Pingvin SoftSound*.

Ako je vaš odgovor potvrdan, nastavite da čitate dalje.

PC RADIO (ovo uslovno čitajte kao „Pingvin Comtrad radio") vraća sjaj staroj i izanđaloj krilatici „štedimo vreme i novac" kroz uobičajenu PC 386 konfiguraciju – ali sa unutrašnjošću obogaćenom neophodnim hardverom i namenski dizajniranim i za radijske potrebe testiranim softverom. Sa ovako koncipiranom mašinom imate mogućnost da bez ikakvih ograničenja program iz minuta u minut „filujete" najavama, špicama, džinglovima, muzikom, zvučnim efektima i reklamama.

Više vam nisu potrebne arhive sa trakama i kasetama, a zaboravićete i na mukotrpno šniranje desetina i desetina tračica sa reklamama i sitnim programskim prilozima. Sva se to od sada nalazi na hard-disku računara, a vreme pristupa tonskom zapisu je rekordnih 14 milisekundi!

Potraga za željenim zapisom koji nameravate da emitujete vrlo je jednostavna: preko duhovito rešenog menija i ugrađene baze podataka, koju možete da proširujete onoliko koliko vam dopušta kapacitet hard diska. Fajlove koje želite da emitujete možete birati pojedinačno ili isprogramirati čak nekoliko sati radio-programa, dopuštajući tako kompjuteru da automatizuje posao za koji bi vam bio potreban ton majstor. Višak studijske tehnike možete slobodno prodati ili iskoristiti za opremanje još jednog tonskog studija sa minimalnim brojem analognih uređaja za snimanje i reprodukciju.

Pingvin SoftSound je doživeo vatreno krštenje u emisiji Zorana Modlije „Modulacije" (neka vrsta tehnožurnala u disk-džokejskoj formi, subotom od 15-18 časova na „Radio Pingvinu"), a nakon prezentacije na majskom Sajmu tehnotronike u Beogradu, definitivno zaživeo na programu najmlađeg beogradskog radija. Od tada sve reklame – a ima ih na stotine – kao i bezbrojni džinglovi, zvučne maske i špice svih emisija idu iz „Pingvinovih" studija isključivo kao semplovi sa kompjutera.

Napominjemo da je cena digitalizovanog radio-kita u Pingvin-Comtrad produkciji niža čak i od cene prosečnog studijskog magnetofona, a kvalitet stereo-reprodukcije odgovara CD standardu. U prvu seriju računara ugrađena su *Sound Blaster 16* i *Wave Blaster* zvučne kartice. Međutim, isti tim je već završio projekat dvostruko jeftinije domaće zvučne kartice, sa još boljim performansama.

Autor kompletnog softvera je 21-godišnji programer **Aleksandar Šušnjar**, iza koga je lepa prošlost (Matematička gimnazija, IS Petnica, dve uzastopne pobede na Oktobarskom kupu, nagrada Vajcmanovog naučnog instituta iz Izraela, pobeda na dve svetske informatičke olimpijade) – i još lepša budućnost: zahvaljujući projektu *Pingvin SoftSound*, COMTRAD je odlučio da finansira Šušnjarev odlazak na studije u Ameriku i pomogne mu da se upiše na čuveni MIT (*Massachusetts Institute of Technology*).

Pingvin SoftSound paket za semplovanje, obradu i reprodukciju zvuka visokih performansi sigurno će se dopasti radio stanicama, TV kućama, privatnim studijima, diskotekama, razglasnim stanicama u školama, preduzećima, hotelima, letovalištima, robnim kućama i tržnim centrima – a bogme i dobrostojećim entuzijastima. Uostalom, detalje ćete saznati ako pozovete **Radio Pingvin (011/697-130)** ili lično posetite ovu kuću i vidite PSS na delu.

NAPRAVI MI MALO MESTA

Hard-drajvovi su svakim danom sve jeftiniji, ali prostor na disku i dalje vredi kao suvo zlato. Podaci se gomilaju, sve je više bit-mapirane grafike, i ono što treba smestiti u memoriju raste mnogo brže nego što se smanjuje cena hardvera. Rešenje koje može poslužiti umesto hardverske dogradnje, off-line skladištenja ili stalnog potkresavanja datoteka jeste transparentna (*on-the-fly*) kompresija podataka.

Kompresija podataka je stara stvar. PKZip i LHARC su samo neki od uobičajenih uslužnih dodataka za arhiviranje datoteka. Oni arhiviraju ili komprimuju datoteke tako da zauzimaju manje prostora, ali tako sažete datoteke bi pre upotrebe morale „raspakovati" odnosno dekomprimovati. Uz pomoć *on-the-fly* kompresije možete udvostručiti efektivni prostor na disku na gotovo nevidljiv način. Jednostavno dobijate iluziju da imate standardni fajl sistem koji zauzima manje mesta.

Ovde će biti prikazana četiri paketa za kompresiju na PC računarima. To su *Double Density 1.0* (Abacus Software), *Stacker 3.0* (Stac Electronics), *SuperStor Pro 1.0* (AddStor) i *XtraDrive 1.0* (Integrated Information Technology). Svi oni rade na principu „presretanja" učitavanja i zapisivanja u sektore. I u proseku postižu kompresiju od oko 2:1.

U ovaj pregled je uključena i najnovija verzija DOS-a, 6.0 – ona sadrži mnoštvo novina, ali upravo *on-the-fly* kompresija kao komponenta operativnog sistema izaziva najviše nedoumica.

DVE PREDNOSTI

Kompresija softverom, naravno, nije isto što i nabavka novog drajva. Performanse će verovatno opasti (mada i tu ima izuzetaka) a na disku ćete imati neobičan format podataka, tako da u slučaju potrebe za oporavkom možete imati problema.

Ipak, kompresija podataka ima dve velike prednosti u odnosu na hardversku dogradnju. Kao prvo, jeftina je. Cena paketa za kompresiju iznosi od 59 – 149 dolara, iako da je i najskuplji od njih znatno jeftiniji od sasvim skromnog hard-drajva. Ako vam oni omogućavaju da dobijete solidne performanse, osigurate integritet podataka i udvostručite efektivni prostor diska od 200 MB, onda ste uštedeli priličnu sumu.

Druga prednost softverske kompresije čini je posebno efikasnom tehnologijom za korisnike *notebook* računara. Naime, nije nimalo praktično ugrađivati dodatni skladišni prostor u prenosive računare, a softverska kompresija eliminiše potrebu za hardverskom dogradnjom.

Ako je softverska kompresija pravo rešenje, ostaje pitanje izbora odgovarajućeg paketa. Prava mera efikasnosti programa za kompresiju svodi se na stepen kompresije i brzinu kompresije i dekompresije – to su dva kriterija na osnovu kojih se testiraju programi za kompresiju. Stepen kompresije jeste funkcija vrste podataka i efikasnosti kompresora, na primer tekstualne datoteke se mogu komprimovati u odnosu 2:1, dok kod bit-mapiranih datoteka taj odnos može da bude i 8:1.

Od brzine kompresije i dekompresije zavisi koliko će rad *on-the-fly* kompresora biti transparentan. Tabela 1 prikazuje brzinu i sposobnost kompresije svih paketa.

Posle ova dva glavna faktora treba razmotriti cenu, lakoću korišćenja, minimalne har-



dverske zahteve, kao i alatke koje paket sadrži (vidi tabelu 2).

TRANSPARENTNA KOMPRESIJA

Kompresija podataka se obavlja tako što se podaci prevode iz jednog seta simbola u drugi, koncizniji niz simbola. *On the-fly* kompresori koriste različite kompresione algoritme, a veći na proizvođača svoju tehniku drži u tajnosti. Međutim, najuobičajeniji algoritmi za univerzalnu kompresiju predstavljaju varijaciju shema zasnovanih na rečniku, kao što su LZ (Lempel Ziv) i njemu srodna shema LZW (Lempel Ziv Welch). Na primer, Stacker koristi kompresioni algoritam zvani LZS (Lempel Ziv Stacker).

Kompresori zasnovani na rečnicima koriste simbole za predstavljanje stringova koji se ponavljaju u nekomprimovanoj verziji inputa. Rečnik za kodiranje mapira ove simbole na stringove koje oni predstavljaju. Kod većine algoritama, dekompresor može potpuno rekonstruisati rečnik za kodiranje iz komprimovanih podataka – dakle, kompresor ne mora da eksplicitno sadrži tabelu za dekodiranje.

LZ i slične tehnike omogućuju brzu kompresiju i dekompresiju, mada je dekompresija obično nešto brža. Za transparentne operacije brzina je od suštinskog značaja.

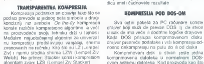
Kao i svaka kompresija koja ne oštećuje podatke, *on-the fly* kompresija radi tako što uklanja redundantnost iz izvornih podataka, dakle ona u velikoj meri zavisi od tipa podataka. Izvorni podaci sa visokim stepenom redundantnosti (npr. bit mape ili uglavnom prazne baze podataka) veoma su pogodni za kompresiju, dok izvršne datoteke, odnosno programi ili već komprimovane arhive nisu. Tekstualne datoteke obično se nalaze negde na sredini.

Da bi iluzija o transparentnoj operaciji bila potpuna, neki *on the fly* kompresori „obmanjuju" operativni sistem, ne saopštavajući im pravu istinu o zauzetom i raspoloživom prostoru na disku. Tačnost procene kompresora o slobodnom prostoru zavisi od toga koliko su podaci koje ćete smeštati pogodni za kompresiju: ako je disk napola popunjen tekstualnim datotekama, dodavanje izvršnih programa ili bit-mapa u drugu polovinu verovatno će za posledicu imati čudnovate rezultate.

KOMPRESIJA POD DOS-OM

Sva četiri paketa za PC računare koriste drajver koji služi da prevari DOS, tj. da stvori utisak da ima veće ili dodatne logičke drajvove. Kada DOS pristupa komprimovanom disku, drajver presreće podatke i vrši kompresiju odnosno dekompresiju na putu do ili od diska.

Komprimovani disk u stvari jeste jedna komprimovana datoteka u normalnom DOS-ovom sistemu datoteka. Kod proizvoda kao što su *Stacker* ili *SuperStor Pro* kompresiona datoteka obično zauzima najveći deo raspoloživog prostora na tvrdom disku. Osim ove datoteke, tu je obično samo jedna kopija datoteke CONFIG.SYS, kompresioni drajver i još neki



Prosečni stepen kompresije i indeks propusnog opsega. Kod kompresije, 2,0 označava da je stepen kompresije 2:1 – kapacitet diska je udvostručen. Testirali smo proizvode koji nude različita podešavanja: optimizovanu brzinu (maksimalna brzina, najslabija kompresija) ili optimizovanu veličinu (maksimalna kompresija, najmanja brzina). Propusni opseg komprimovanog diska je meren u odnosu na performanse nekomprimovanog diska na mašini Compaq Deskpro 386/33L. Veće cifre znače bolji rezultat.

Tabela 1 Testovi brzine

	Tekst-procesori		Baze podataka		Grafika	
	stepen komp	prop opseg	stepen komp	prop opseg	stepen komp	prop opseg
DoubleDensity	2 00	0 36	2 00	0 26	2 00	0 37
Stacker	2 06	1 05	1 92	0 95	2 10	1 02
SuperStor Pro	1 50	0 62	1 40	0 64	1 30	0 56
XtraDrive	1 70	0 73	1 70	0 62	1 72	0 73

Stepen kompresije i indeks propusnog opsega za tri osnovne kategorije softvera. Kod svakog tipa aplikacije, stepen kompresije se odnosi na tipične podatke. Indeks propusnog opsega za tekst-procesore i baze podataka predstavljaju kombinaciju BYTE-ovih aplikativnih testova i testove na mašinskom nivou (vidi „Merenje kompresije"). Veće cifre znače bolji rezultat.

drajveri u normalnom prostoru za datoteke.

Potencijalni problem je to što kompresor zahteva dodatno slovo za logički drajv koje se odnosi na komprimovanu datoteku. Ukoliko paket ne sadrži drajver koji sam zamenjuje slova za označavanje drajvova, moraćete podesiti sve staze (path) da bi bio moguć rad sa novim drajvom.

Kompresija nastaje tako što se izvestan broj sektora zapisanih na disk komprimuje u manji broj sektora u kompresionoj datoteci. Osim što obavljaju kompresiju *on-the-fly* kompresori efikasnije koriste i prostor na disku. DOS datotekama dodeljuje prostor u klasterima, gde se svaki klaster sastoji od 2 do 32 512-bitnih sektora. Klaster je minimalna jedinica koju DOS dodeljuje – bez obzira koliko u nekoj datoteci ima podataka, svaka datoteka se mora sastojati od celog broja sektora. Obično hard-drajvovi imaju klastere od 4 KB, što znači da DOS troši dodatna 3 KB prostora da bi smestio datoteku od 1 KB. Pošto se kod kompresora sva interakcija sa DOS-om odvija unutar jedne normalne datoteke, kompresor može racionalnije dodeljivati prostor.

Metod kompresije sa jednom datotekom zasnovanom na sektorima ima izvesne nedostatke. Da bi se pročitao samo jedan bajt, kompresor mora dekomprimovati ceo fizički sektor. Da bi efikasno radio, kompresor mora imati prostrane dekompresione bafere. Osim toga, kompresor mora sam vršiti mapiranje logičkih klastera u fizičke sektore. Svo to čitanje i dekompresija opterećuju mikroprocesor i troše sistemski RAM i vreme za pristupanje disku. Na brzom računaru sa brzim drajvovima, problem brzine se ne postavlja. Srećom, većina programa za kompresiju ima ugrađeno keširanje i ček pruža bolje performanse od nekomprimovanog diska.

Tabela 2 Transparentni (*on-the-fly*) kompresori diska

	DoubleDensity 1 0	Stacker 3 0 Tehnology	SuperStor Pro 1 0	XtraDrive 1 0
Cena	$79 95	$149	$149	$88
Operativni sistem	MS DOS 3 2+	MS DOS 3 2+	MS DOS 3 2+	MS DOS 3 0-
Min memorije	512 KB	640 KB	640 KB	512 KB
Na principu datoteka/sektora	Sektor	Sektor	Sektor	Sektor
Lozinka	Da	Da	Da	Ne
Izbor stepena komp	Ne	Da	Da	Da
Samoraspakivanje datoteka	Ne	Sa flopija	Sa flopija	Ne
De instalacija	Da	Da	Da	Da
ALATKE				
Defragmentor	Ne	Da	Da	Da
Otkriva/ispravlja greške	Ne	Da	Da	Da
Ostalo	Nema	Prenosivi medij Windows alatke JPEG Workshop	Prenosivi medij Windows	Opciono alatke keš

Svi kompresori za PC rade na principu sektora (tj. komprimuju ceo volumen)

Korisnici *Windows*-a moraju biti oprezni: ni jedan od četiri predstavljena programa ne izlazi lako na kraj sa stalnim *swap* datotekama iz *Windowsa*. Pri instalaciji ovih paketa morali smo preći na privremenu *swap* datoteku ili formirati nekomprimovanu particiju za stalnu *swap* datoteku.

Pad sistema je mnogo ozbiljniji problem kod komprimovanog nego kod nekomprimovanog diska, jer se može nepovratno izgubiti više podataka. Međutim, ako redovno vršite *backup* diska, ne bi trebalo da bude razloga za zabrinutost. U svakom slučaju, treba da napravite kompletan *backup* pre no što započnete kompresiju.

DOUBLE DENSITY 1.0

Program *DoubleDensity* kompanije „Abacus Software" je jednostavna i jeftina alatka za kompresiju. Predviđen je za DOS i, mada se izvršava i pod *Windows*-om, ne raspolaže nekim tipičnim *Windows* opcijama koje se mogu naći u drugim paketima.

Da biste instalirali *DoubleDensity*, treba prvo da otkucate registracioni broj flopi-diska kako biste „probili" zaštitu i dobili ovlašćenje za kopiranje. Zatim birate instalacioni poddirektorij i disk koji želite da arhivirate (maksimalan kapacitet za *DoubleDensity* iznosi 250MB). Korisniku se nudi i opcija da odabere lozinku (do osam karaktera) i tako zaštiti svoje podatke. Pre nego što započne proces kompresije, *DoubleDensity* koristi **chdisk** za skeniranje diska u potrazi za izgubljenim klasterima.

Pri instalaciji, *DoubleDensity* automatski modifikuje CONFIG.SYS datoteku. Ako želite da se učita u visoku memoriju, morate sami izvršiti izvesne modifikacije. Drajver zauzima 47KB RAM-a.

Instalacija vam ne nudi mogućnost da odaberete standardni (*default*) stepen kompresije. *DoubleDensity* povećava kapacitet tako što kreira skrivenu datoteku zvanu D_D_.VOL koja zauzima sav raspoloživi prostor za skladištenje. Drajveri kreiraju novi komprimovani disk koji je gotovo dvostruko veći od originalnog. Kada se završi proces instalacije, sav softver koji se nalazio, na primer, na drajvu D naći će se na većem drajvu E, a korisnik sam treba da promeni stazu (path).

Što se alatki tiče, *DoubleDensity* je najsiromašniji od četiri paketa. Njegova komanda **chkddv** je slična DOS-ovom DIR, ali pruža i dodatne informacije o kompresiji i raspoloživom prostoru. **ddexit** vas obaveštava da li je disk *DoubleDensity* volumen ili nije, dok **dall**

DOUBLESPACE I DOS 6.0

DOS 6.0 ima kompresiju podataka ugrađenu u operativni sistem u obliku alatke koja se zove *Doublespace*. U stvari, *Doublespace* je predmet sudskog spora koji je kompanija „Stac Electronics" pokrenula protiv „Microsoft"-a, tvrdeći da *Doublespace* koristi njihovu tehnologiju kompresije.

Sa stanovišta korisnika, DOS 6.0 obavlja kompresiju slično kao *Stacker* i *SuperStor Pro*. Nakon instalacije paketa DOS 6.0 otkucate **dblspace**. *Doublespace* kreira takozvani CVF (*compressed volume file*) i instalira drajver koji „obmanjuje" DOS, navodeći ga da CVF smatra logičkim drajvom. Osnovna razlika između *Doublespace*-a i drugih kompresora za PC je u tome što kod DOS-a kompresioni drajver predstavlja kernel datoteku operativnog sistema – njene rutine postaju deo MS-DOS operativnog sistema u času kada se podiže sistem.

Kada instalirate *Doublespace*, možete da birate između dve opcije: da arhivirate sve postojeće datoteke tamo gde jesu ili da kreirate prazan kompresioni logički drajv. U prvom slučaju, *Doublespace* skenira celu particiju, arhivira nekomprimovane datoteke, premešta ih u CVF i briše originalne datoteke iz regularne particije.

DOS 6.0 sadrži i MRCI (*Microsoft Real Time Compression Interface*) „Mersi", kako se ovaj akronim čita, omogućava da aplikacija od operativnog sistema dobije podatak o prisustvu kompresionog servera. Ukoliko taj server postoji, aplikacija može pozvati serverske rutine za kompresiju, odnosno dekompresiju. Na primer, ukoliko neki telekomunikacioni program treba da pošalje veliku količinu podataka u mrežu, on može prvo prosleđiti podatke kompresionom serveru, a zatim ih poslati u mrežu u arhiviranom obliku.

DOS 6.0 raspolaže LZ (Lempel Ziv) kompresionim serverom. To je istovremeno kompresiona mašina pomoću koje *Doublespace* uglavnom radi. Testiranje obavljeno sa beta verzijom je pokazalo da DOS 6.0 daje stepen kompresije čija je vrednost negde u sredini u odnosu na ostale DOS pakete, dok je postignut propusni opseg u rangu najboljih rezultata. Naravno, pravi pokazatelji će se znati tek kada se testira konačna verzija.

daje listu svih *DoubleDensity* volumena koji se nalaze na sistemu.

DoubleDensity nije skup proizvod ($79.95) i relativno je jednostavan za upotrebu. Od svih prikazanih paketa postiže najbolju kompresiju, ali, na žalost, on je i najsporiji od svih, a nedostaju mu i alatke za procenu raspoloživosti diska na osnovu tipa datoteke, kao i za korekciju grešaka. Zbog svega toga *DoubleDensity* nije kompresor koji bi se mogao preporučiti za upotrebu.

STACKER 3.0

Stacker za *Windows* i za DOS, verzija 3.0 (proizvođač „Stac Electronics") je gotovo transparentan kompresor, a brzina mu je takva da uglavnom ne dovodi do osetnog pada performansi. Ima interfejs lak za korišćenje, kako za DOS tako i za *Windows*. Sa cenom od $149 nije baš jeftin, ali je po brzini i praktičnosti apsolutno bez premca.

Stacker ima dva različita metoda instalacije: jednu za DOS a drugu za *Windows*. Pri instalaciji možete podesiti *Stacker* da optimizuje kompresiju i brzinu, ili se pak možete odlučiti za umerene vrednosti.

Stacker kreira STACVOL datoteku. Ova nevidljiva datoteka sadrži sve komprimovane datoteke u „Stacked" drajvu. Pri testiranju je STACVOL datoteka zauzela 10 378 192 bajta, ostavljajući oko 100 KB *Stackeru* kao privremeni prostor za skladištenje. Nema značajnih restrikcija u pogledu obima – nekomprimovani disk može imati do jednog GB, a jedino ograničenje je 10 *Stacker* drajvova po sistemu.

Dva *Stacker* drajvera mapiraju slovo za oznaku drajva u komprimovanu datoteku i zamenjuju oznaku drajvova. Ako arhivirate drajv C, komprimovani drajv će i dalje biti C, dok će nekomprimovani postati D. Drajveri su u stanju da registruju stalnu *swap* datoteku iz *Windows*-a i mogu da se konfigurišu tako da se *swap* datoteka zaobiđe svaki put kada se startuje sistem. Drajveri koriste od 36 KB do 64 KB RAM-a. Mogu se učitati u visoku memoriju ili se *Stacker* može konfigurisati tako da koristi EMS, čime se smanjuje potreba za sistemskim RAM-om.

Stackerov standardni (*default*) stepen kompresije iznosi 2:1, ali se može podešavati, do maksimuma od 16:1. Paket sadrži i verziju programa *Norton Speed Disk* koji služi za defragmentaciju datoteka. Kada *Speed Disk* završi svoj deo posla, *setup* program skenira celi arhivirani disk i traži loše sektore.

Lepa novina je to što *Stacker* raspolaže *Windows* opcijama. Nakon arhiviranja diska možete kreirati *Stacker Program Group*, čime dobijate ikonice za *Stacker Tuner*, *setup* opcije, *readme* datoteku, kao i za alatku po imenu *Stackometer*. *Stackometer* ima instrumente pomoću kojih daje sumarne podatke o stepenu kompresije arhiviranog diska. Linijski grafikon pokazuje koliko je na disku potrebno prostora za nearhiviranu datoteku i koliko ista datoteka zauzima na arhiviranom disku. *Disk Space* pokazuje zauzetost i raspoloživi kapacitet arhiviranog diska, a može se dobiti i grafički prikaz fragmentacionih nivoa diska.

Međutim, nisu zapostavljeni ni korisnici DOS-a. Sve *Stackerove* alatke se mogu pozivati i iz komandne linije. Na primer, komandom **report** dobijate detaljne informacije o korišćenju diska – listu svih datoteka na odabranom drajvu i njihov trenutni stepen kompresije. Ugrađen je i optimizator pomoću kojeg možete promeniti raspored datoteka na disku. Opcija **quick optimizer** raspoređuje samo fragmentirane datoteke, dok **full optimizer** optimizuje celokupni disk.

KAKO SE MERI KOMPRESIJA

Brzina je merena pomoću kombinacije aplikativnih testova brzine (*benchmark*) i testova na mašinskom nivou. Startovani su BYTE-ovi testovi brzine za tekst-procesora i baze podataka. Za testiranje „sirove" brzine startovane su direktne i sekvencijalne *input/output* operacije sa različitim podacima koji reprezentuju najčešće tipove datoteka: tekstualne, izvršne datoteke, odnosno programa, bit mapirane grafike, baze podataka i arhivirane podatke.

Brojevi dati u tabeli predstavljaju prosečan stepen kompresije za sve tipova datoteka, odnosno ukupni indeks propusnog opsega. Podaci o propusnom opsegu se dobiju na osnovu odnose između propusnog opsega određenog paketa za kompresiju i propusnog opsega koji sistem postiže bez kompresije.

Testiranje je vršeno na računaru Compaq Deskpro 386/33L.

Stacker Tuner omogućava podešavanje odnosa između brzine *Stackera* i gustine kompresije. Bez obzira za koju opciju se odlučite, *Stacker* postiže zavidnu brzinu. Kada zahtevate optimalnu brzinu, *Stacker* je brži od nearhiviranog diska. Stepen kompresije je manji od onog koji postiže *DoubleDensity*, ali je, kada se odabere optimalna kompresija, veći nego kod ostalih paketa.

Veoma praktična osobina *Stackera* je mogućnost arhiviranja flop-diska i drugih prenosivih medija. Komprimovan prenosivi disk može se koristiti na bilo kojem računaru, čak i ako nema instaliran *Stacker*, i na njemu čitati i zapisivati podatke u memoriju. Naime, komprimovani disk ili kertridž ima dve vidljive datoteke – README.STC i STACKER.EXE (odnosno program *Stacker Anywhere*). Pokretanjem ovog programa dobijate mogućnost da pristupite podacima na prenosivom disku.

Sve u svemu, *Stacker* je brz, postiže dobru kompresiju i zgodan je za upotrebu. Poseduje alatke koje pružaju precizne podatke o potrošnji prostora na tvrdom disku i o stepenu kompresije, a ima i funkcije za korekciju grešaka. Konačno, od četiri prikazana programa, *Stacker* ima najbolju dokumentaciju.

SUPERSTOR PRO 1.0

Ukoliko se *Stacker* uzme za referencu sa kojom poredimo ostale DOS kompresore, onda *SuperStor Pro* (proizvod kompanije „AddStor") dolazi odmah posle njega. Prodaje se po istoj ceni kao i *Stacker* ($149) i rad pod *Windows*-om, ali je nešto sporiji, ima manju moć kompresije i nije tako fleksibilan kao *Stacker*.

Pre instalacije paketa *SuperStor Pro* morate isključiti rezidentne programe (npr. EMM386 ili Compaq CEMM). Instalacioni program kopira drajver i uslužne programe na disk, modifikujući pri tom CONFIG.SYS tako da se pri svakom podizanju sistema startuje drajver.

Instalacija se nastavlja tako što se startuje glavni korisnički program paketa *SuperStor Pro*, SSTOR. Ovaj program pruža pregled rezultata kompresije, dajući podatke o arhiviranom i nearhiviranom disku. Zatim se disk priprema za kompresiju. Tokom pripreme, program prepoznaje stalnu *swap* datoteku *Windows*-a i nudi da sačuva *swap* datoteku na rezervisanoj particiji ili pak da je izbriše.

Zatim vas SSTOR pita da li želite da arhivirate celi disk ili samo neki njegov deo. Možete birati i stepen kompresije: od 2:1 do 8:1. Predviđena (*default*) vrednost je 2:1.

Ako se odlučite da komprimujete particiju sa koje se ne diže sistem, morate kreirati drugi drajv koji se softverski ugrađuje (*mount*) u sistem. Uputstvo za ovakvu instalaciju nije baš najjasnije i za podacima treba tragati u različitim poglavljima dokumentacije. Da bi se arhiviralo 10 MB jedne particije drajva D, prvo se mora pripremiti disk, tj. mora se softverski ugraditi u sistem. Zatim se skrivenoj datoteci mora dati jedinstveno ime. Da bi se komprimovana particija automatski startovala, mora se podesiti datoteka AUTOEXEC.BAT.

Kao i *Stacker*, *SuperStor Pro* može promeniti odredišni drajv, tako da arhivirani drajv zadrži svoju raniju oznaku – na primer, nearhivirani drajv D postaje arhivirani drajv D.

Drajver zauzima 52KB RAM-a pri običnoj instalaciji. Ukoliko se učita u visoku ili proširenu memoriju, potreba za RAM-om se smanjuje na svega 10KB.

SuperStor Pro ima sposobnost arhiviranja flopi-diska, kao i drugih prenosivih medija. Pomoću opcija iz SSTOR menija ugrađuje se i arhivira odabrani disk. Program SSTOR omogućava da datoteku na prenosivom disku proglasite *read-only* ili *read-write* datotekom. *Read-write* arhivirani disk sadrži drajvere i za DOS i za *Windows*, kao i *Windows* DLL (*Dynamic Link Library*). Kada prenosivi disk odnesete na drugu mašinu, treba samo da pokrenete odgovarajući drajver, zavisno od toga pod kojim operativnim sistemom računar radi.

SuperStor Pro ima više uslužnih programa, a gotovo svi se nalaze u *shell*-u SSUTIL. SSUTIL pruža čitav niz osnovnih funkcija za održavanje. Funkcija za analizu je slična onoj koju ima CHKDSK: analizira i otklanja oštećenja na arhiviranom disku. *Tune up* komponenta vrši fragmentarnu analizu i obavlja punu ili delimičnu defragmentaciju arhivirane datoteke. SSUTIL može ponovno komprimovati već arhivirane podatke, povećavajući kompresiju za oko 25% (s tim što se tada kompresija vrši sporije). Konačno, korisnik može podesiti veličinu *SuperStor Pro* komprimovane particije. Osim uslužnih programa koje sadrži SSUTIL, postoji i SDIR koji se jednostavno poziva iz komandne linije i daje listu datoteka na ugrađenom drajvu, njihovu veličinu, stepen kompresije, vreme i datum.

Neke *SuperStor Pro* alatke se dobijaju preko dobro dizajniranog *Windows* interfejse. *Windows* aplikacija SSWIN PRO je informativnija od slične koju nudi *Stacker*. Uz *SuperStor Pro* se dobija i program *JPEG Workshop*, lepa *Windows* aplikacija za kompresiju slika. *JPEG Workshop* nije integrisan u *SuperStor Pro*, već se izvršava kao zaseban program.

Na žalost, *SuperStor Pro* je najslabiji u

Nastavlja se na strani 51

UKRATKO

Šta su kompresori?

Softverski paketi koji povećavaju skladišni prostor na disku, obavljajući automatsku kompresiju/dekompresiju podataka.

Prednosti

Visok stepen kompresije i zavidna brzina. Kompresija je jeftina alternativa za hardversku dogradnju.

Mane

Oporavak podataka može biti problem, posebno kod programa *XtraDrive*. Pri setup-u potrebno podešavanje konfiguracionih datoteka.

Preporuka

Stacker

NOVI CLIPPER

Clipper od skora ima ne samo novo odelo, već i novo ime: sada se zove _CA-Clipper_, jer je _Nantucket_ otkupila firma _Computer Associates_. Ako je verovati vestima, _Clipper_ će tamo imati jako lep dom. U narednom tekstu videćemo kako izgleda najnovija verzija _Clipper_-a, potekla upravo iz ove kuće.

Nenad Batoćanin

Kada je izašla verzija 5.0, *Clipper* je definitivno prestao da bude *Clipper*: radilo se o potpuno novom proizvodu, kojem je kompatibilnost sa verzijom *Summer'87* bila tek jedna od poželjnih osobina. U suštini, ova kompatibilnost je veliki balast: arhitekti verzije 5.0 su imali veoma sužen prostor. Treba reći da su oni i pored toga uspeli da konstruišu zaista izuzetnu građevinu. Sa druge strane, kompatibilnost sa strahovito rasprostranjenim *Clipper S'87* je odličan poslovni potez, time se u startu odbija ogroman broj korisnika.

Jedna od osnovnih karakteristika *Clipper*-a (svih verzija) je otvorenost arhitekture. To znači da programer može da izmeni deo sistema po svojim željama, tako je, na primer, moguće izmeniti sistem za obradu grešaka. Međutim, tek *Clipper 5.0* je dobio zaista otvorenu konstrukciju. Izuzetan preprocesor i otvorenost ostalih podsistema omogućava veoma jednostavno širenje i modifikovanje osnovnog programskog jezika. Ova dobra osobina se pokazala mnogo pre izlaska verzije 5.2: veliki broj nezavisnih proizvođača je svoja „dodatke" za *Clipper* uobličio u dodatne naredbe, koje se potpuno ravnopravno koriste sa „fabričkim" naredbama. To je gotovo sasvim otklonilo problem kompatibilnosti narednih verzija takvog softvera i *Clipper*-a.

Ako je verovati dokumentaciji, *Clipper 5.2* je potpuno kompatibilan sa prethodnim verzijama. Što je posebno značajno, u dokumentaciji (i u **NG** bazi podataka koja ide uz *Clipper*) je veoma jasno naglašeno šta je sve menjano, a poseban deo se odnosi na opis tih izmena. Pre prelaska na verziju 5.2 svakako proučite ovaj deo, kako biste izbegli nepotrebne komplikacije. Vaši programi bi trebalo da proradе sa novom verzijom bez većih izmena, ipak, kao što znate, to u praksi ne ide uvek glatko.

Najznačajnija novina u verziji 5.2 je korišćenje RDD-a – drajvera koji omogućavaju pristup raznim formatima podataka. Većina novih formata takođe koristi **DBF** datoteke za smeštanje podataka, a razlikuje se u indeksima. Da počnemo redom.

INDEKSI

Naredbe za indeksiranje su u novoj verziji pretrpele veoma značajne izmene. Kao prvo, uvedena je opcija **FOR** koja omogućava da se kreira indeks samo za deo datoteka. Na primer, naredba

INDEX ON Upper(Ime) TO IndIme ;
FOR "BEOGRAD" $ Adresa

kreira indeksnu datoteku **IndIme** po izrazu **Upper(Ime)**, ali samo za slogove koji u polju **Adresa** sadrže string "**BEOGRAD**". Ostali slogovi neće biti „vidljivi" dok je aktivan ovaj indeks. Iako da se ova naredba može zgodno upotrebiti za neku vrstu filtriranja podataka. Na primer, ako posle prethodne naredbe pozovemo proceduru **Browse()** za tabelarni pregled, videće se samo izabrani slogovi. Ova opcija lako vrlo uspešno može zameniti sporu naredbu **SET FILTER** za izdvajanje dela datoteke.

Opcija **FOR** nije jedini način za kreiranje parcijalnih indeksa: može se specificirati i proizvoljan deo **DBF** datoteke. Zada se opseg datoteke za koju se kreira indeks sa **ALL** (cela datoteka), **REST** (od tekućeg sloga do kraja), **NEXT** (određen broj slogova od tekućeg), **RECORD** (određeni slog). Zatim se od zadatog skupa slogova izdvajaju željeni opcijama **FOR** ili **WHILE**. Na primer, sledeće naredbe kreiraju indeks za sve slogove posle sloga sa rednim brojem 100, sve dok je polje **PB = "11000**

GO 100
INDEX ON Upper(Ime) TO IndIme WHILE PB = = "11000

Veoma zgodan dodatak su i opcije **EVAL** i **EVERY**. Ove opcije omogućavaju da se određena korisnička funkcija izvršava prilikom kreiranja indeksa. Na primer, ako napišete

INDEX ON Broj TO IndBroj EVAL {|| QOut(RecNo()), .T.}

prilikom indeksiranja će se ispisivati redni broj sloga koji se obrađuje. Ako vam se čini da isti posao obavlja sledeća naredba

INDEX ON Broj + WriteFunc() TO IndBroj

prevarili ste se: u drugom slučaju se funkcija **Func()** uključuje u indeksni izraz, pa će biti pozivana i za vreme **računanja** indeksa, što sigurno niste želeli.

Zadata funkcija može bitno usporiti kreiranje indeksa u slučaju većih datoteka (funkcija se poziva kod kreiranja indeksa za svaki slog, znači ako datoteka ima 10000 slogova, toliko puta će biti pozvana). Zato postoji opcija **EVERY** koja određuje na koliko slogova će biti pozvana zadata funkcija. Opcije **EVAL** i **EVERY** sada ima i naredba **REINDEX**.

U *Clipper*-u 5.2 za kreiranje indeksa sa obrnutim redosledom u odnosu na normalni ne mora se koristiti funkcija **Descend()** u indeksnom izrazu, dovoljno je upotrebiti opciju **DESCENDING** ili **ASCENDING**. Na primer

INDEX ON Broj TO BrInd DESCENDING

kreira indeks na osnovu vrednosti polja **Broj**, ali u opadajućem redosledu.

DRAJVERI ZA PRISTUP BAZI PODATAKA

Većina *DBMS* paketa za *MS DOS* koristi **DBF** datoteke za smeštanje podataka i to je nasleđeno od *dBASE*-a. **DBF** standard je veoma jednostavan za realizaciju, ali ima i priličan broj nedostataka. Zato su svi noviji programi na neki način dograđivali standardni **DBF**. Na primer, *Clipper* je umesto originalnih **NDX** indeksa uveo znatno bolje **NTX** indekse, *Fox Pro* je uveo još kvalitetnije **CDX** indekse, i slično. Neki paketi (*Paradox*) uopšte ne koriste **DBF** format, već svoj poseban format zapisa. Kao što vidite, siguran put u haos.

Slika 1

	NTX	NDX	MDX	CDX	DBPX
Zaključavanje vise slogova	Da	Da	Da	Da	Ne
Tagovi	Da	Da	Da	Da	-
Broj tagova po indeksnoj datoteci	1	1	47	50	
Broj indeksnih datoteka po radnoj zoni	15	15	15	15	
Uslovni indeksi	Da	Ne	Da	Da	Ne
Parcijalni indeksi	Da	Ne	Ne	Da	Ne
EVAL i EVERY	Da	Ne	Ne	Da	Ne

RDD Drajveri koji se dobijaju uz Clipper 5.2

Slika 2

	NTX	CDX
Indeksiranje	45.00	63.20
Traženje proizvoljnog sloga	1.48	2.58
Sekvencijalna obrada sa indeksom	4.34	4.39
Izmena izraza u indeksu	63.49	56.08

Performanse DBFNTX i DBFCDX drajvera

Sa druge strane, dosta je nerealno očekivati da korisnik ima sve programe urađene samo pomoću jednog alata i samim tim sve podatke u istom formatu. Ako je reč o korisniku koji koristi veći broj poslovnih programa, sasvim je izvesno da će vaš program morati da komunicira sa drugim programima, pa znači i da koristi podatke zapisane u drugim formatima. Svi bolji programi podržavaju neku konverziju podataka, ako ništa drugo, prvo se **DBF** konvertuje u **ASCII**, a zatim se to konvertuje u ciljni format. Bez obzira kolika je polazna datoteka, ovo je operacija koja traje. Zato je veoma prirodno postaviti pitanje: a zašto ne koristiti **direktno** podatke u drugom formatu?

Odgovor je uglavnom u tehničkim razlozima: ako bi program dozvoljavao pristup raznim formatima, sistem za komunikaciju sa bazom podataka bi morao biti znatno komplikovaniji, zatim, tu je čitav niz pitanja oko veza na relaciji program-podaci. Doduše, većina savremenih programa podržava konverziju svog formata u druge i obrnuto, ali direktno korišćenje drugih formata je retko.

Clipper 5.0 je imao pripremljeno rešenje za ovaj problem. U skladu sa principom „otvorene arhitekture" čitav sistem za pristup bazi podataka je bio izdvojena celina predviđena za izmene. To se u dokumentaciji *Clipper*-a naziva **RDD** (*replaceable database driver*). Na žalost, to nije bilo dokumentovano u verziji 5.0. Bez obzira na to (kao što to uvek biva) na tržištu se vrlo brzo pojavilo nekoliko programa koji su koristili ovu mogućnost i omogućavali *Clipper*-u da pristupa podacima u drugim formatima (*FlexFile*, *SIX driver*, ...). U najnovijoj verziji je postojanje **RDD**-a odkumentovano, ali i dalje se detalji o tome moraju tražiti na poseban zahtev. U katalogu **INCLUDE** *Clipper*-a 5.2 može se naći datoteka **RDD API** koja je, u stvari, heder za potencijalni **RDD** drajver pisan u *C*-u. Na žalost, samom analizom ove datoteke se ne može baš previše zaključiti. Ostaje nam samo da čekamo narednu verziju, a do tada možemo koristiti gotove drajvere koji se isporučuju uz *Clipper 5.2*.

Sve u svemu, ideja **RDD**-a je vrlo prosta: funkcije za pristup podacima su nezavisne od naredbi i funkcija odslupnih korisniku, pa jednostavnom zamenom drajvera program bez ikakvih izmena može koristiti drugi format podataka. U paketu se dobijaju četiri drajvera: **DBFNTX** (standardni *Clipper* format), **DBFNDX** (*dBASE III*), **DBFCDX** (*FoxPro*), **DBFMDX** (*dBASE IV*), **DBPX** (*Paradox*). Svaki drajver je u stvari jedna biblioteka koja se po potrebi uključuje u **EXE** program. Ako koristite *Clipper* na standardni način, podrazumeva se da se koristi **DBFNTX** drajver i to je (sa manjim izmenama, kao što je uslovno indeksiranje) poznati sistem **DBF** datoteka i **NTX** indeksa. Znači, svi stari programi bi trebalo da rade bez ikakvih problema. Međutim, ako na početku programa stavite sledeću naredbu

RddSetDefault ("DBFCDX")

onda ći podrazumevani drajver biti **DBFCDX** i vaš program će (skoro) bez ikakvih izmena koristiti *FoxPro* kompatibilne indekse i memo polja! Što je još lepše, u okviru istog programa možete paralelno ko-

nstiti više drajvera – potrebno je samo upotrebiti opciju VIA. Na primer:

USE Artikli VIA "DBFNTX" INDEX Art_Naz NEW
USE Partneri VIA "DBFCDX" INDEX Partneri NEW

otvara **DBF** datoteku **ARTIKLI** za korišćenje sa standardnim **DBFNTX** drajverom, a datoteku **PARTNERI** sa *FoxPro* kompatibilnim (.**CDX**) indeksom. Naredbe za obradu u programu su dalje potpuno iste kao da se koristi standardni drajver.

Svakako najinteresantniji drajveri su **DBFNTX** i **DBFCDX**. Ostali drajveri (**DBFNDX** i **DBFMDX**) obezbeđuju suviše niske performanse da bi bili posebno interesantni. Njihova jedina primena može biti verovatno samo u slučajevima kada treba koristiti podatke iz drugih programa. Drajver **DBPX** je kompatibilan sa *Paradox*-om 3.5 i može biti interesantan, ali treba reći da se ovaj format ne uklapa u *xBase* filozofiju, tako da bi za njegovo korišćenje bilo potrebno značajno menjati programe. Na slici 1. je data tabela drajvera koji se dobijaju uz *Clipper* 5.2 i nekih njihovih osobina.

Na slici se vidi dosta pojmova nepoznatih u prethodnim verzijama *Clipper*-a: tag, parcijalni indeks... U dokumentaciji se veoma potenciraju ti novi pojmovi, očigledno se predviđa da će *Clipper* pristupati raznim bazama podataka, od kojih neke neće biti ni malo blizu **DBF** standarda. Tada će neke naredbe i funkcije izgubiti svoj početni smisao (na primer, *Paradox* čuva podatke u povezanoj listi, pa termin „broj sloga" tu ne znači previše). Iz tog razloga se u dokumentaciji za *Clipper* 5.2 nalaze i definicije novih pojmova, pa čak i mali rečnik koji treba da omogući lakše snalaženje. Treba reći da je čitava stvar pozitivna, ali da je donekle iskomplikovala objašnjenje postojećih naredbi i funkcija.

FOX I CLIPPER

Na prvi pogled, **DBFCDX** drajver je najinteresantniji od svih priloženih, i to iz više razloga. Prvo, za podatke se koristi standardna **DBF** datoteka, pa nije potrebno konvertovati stare podatke, osim ako postoje memo polja. Dalje, **CDX** indeksi su potencijalno bolji od **NTX** indeksa. Pri kreiranju **CDX** indeksa se vrši kompresija, tako da se ne upisuje uvek vrednost celog indeksnog izraza, već samo razlika u odnosu na prethodnu vrednost. Time se štedi na memorijskom prostoru, što je veliki dobitak pri radu u mreži i uopšte na sistemima gde je pristup disku usko grlo.

U stvari, ekvivalent **NTX** indeksa u *FoxPro*-u su **IDX** indeksi. **CDX** indeksi su sastavljeni iz više prostih **IDX** indeksa. Time je omogućeno da se svi indeksi koji se odnose na jednu **DBF** datoteku drže u jednoj datoteci, tako se izbegavaju problemi oko neažurnih indeksa i smanjuje se broj istovremeno otvorenih datoteka (umesto više indeksnih datoteka koristi se samo jedna sa svim indeksima), što nije nikako za zanemarivanje, naročito pri radu u mreži.

Pri kreiranju indeksa (u novoj terminologiji „Order") svakom indeksu se može dodeliti identifikator koji kasnije služi kao oznaka za aktiviranje: umesto da zadajete aktiviranje indeksa broj 5, aktivirate indeks Adresa, što je mnogo preglednije. Ovo je dodatak za naredbu **SET ORDER TO**, koja sad ima mogućnost zadavanja i simboličkog imena za indeks. Ovo se može koristiti i za standardne **NTX** indekse. Na primer:

USE Adresar
INDEX ON Upper(a_naziv) TAG AdrNaz TO Adr_Naz
INDEX ON a_id TAG AdrID TO Adr_ID
...
USE Adresar INDEX Adr_Naz, Adr_ID NEW
SET ORDER TO TAG AdrID

Prethodne naredbe kreiraju standardne **NTX** indekse za datoteku **Adresar** po dva izraza, ali pri tom se svakom indeksu dodeljuje simboličko ime (Tag), pa se kasnije umesto **SET ORDER TO 2** može mnogo preglednije pisati **SET ORDER TO TAG AdrID**. Ovo simboličko ime se upisuje u indeksnu datoteku, tako da je dostupno uvek pri kasnijem korišćenju indeksa. U prvobitnim verzijama *Clipper*-a je postojala greška, tako da se simboličko ime uz opciju **TAG** moralo uvek pisati velikim slovima.

Kao što smo već videli, **CDX** indeksne datoteke mogu sadržati više „običnih" indeksa. Evo kako to izgleda:

USE Adresar
INDEX ON Upper(a_naziv) TAG AdrNaz TO Adresar
INDEX ON a_id TAG AdrID TO Adresar
...
USE Adresar INDEX Adresar NEW
SET ORDER TO TAG AdrID

Datoteka **ADRESAR.CDX** sadrži indekse po nazivu i po identifikacionom broju. Kada se otvara **DBF** datoteka, otvara se samo jedna indeksna datoteka i ona u sebi sadrži sve definisane indekse. Izbor aktivnog indeksa kao i u standardnom slučaju obavlja naredba **SET ORDER**.

Clipper 5.2 je obogaćen čitavim nizom funkcija koje se odnose na rad sa ovakvim složenim indeksnim datotekama (*Compound Index*): kreiranje novih indeksa, brisanje itd. Treba napomenuti da se prilikom ponovnog indeksiranja po indeksu koji je već u **CDX** datoteci neće prethodno zbrisati prostor koji zauzima star indeks, već će se novi indeks dodati na kraj datoteke. To znači da će indeksna datoteka uvek rasti kada se upotrebi naredba **INDEX**, bez obzira da li se tom naredbom generiše novi ili već postojeći indeks. Zato za reindeksiranje bar povremeno ne treba koristiti **SET INDEX**, već **REINDEX**, koji vrši prepakivanje cele indeksne datoteke.

Drajver **DBFCDX** ima još jednu veoma lepu osobinu: bolji rad sa memo-poljima. U jednom od ranijih brojeva „Računara" bilo je reči o nedostacima standardnih *Clipper*-ovih memo polja. Ukratko, najveći nedostatak je previše neiskorišćenog prostora. **DBT** datoteka raste u blokovima od 512 bajtova, što je veoma gruba „rezolucija": ako imate memo-polje dužine 1 bajt, ono će takođe alocirati 512 bajtova! **DBFCDX** drajver alocira memoriju za memo polja u datoteci sa ekstenzijom **FPT**, i to u blokovima od 64 bajta, što je neuporedivo bolje: ako izvršite jednostavnu konverziju **DBT** u **FPT**, verovatno ćete dobiti najmanje 20-30% slobodnog prostora.

Ova memo-polja se razlikuju od standardnih i po tome što mogu sadržati sve simbole, pa i kontrolne simbole. **Chr(26)** se u **DBT** memo poljima koristio za oznaku kraja, pa nije mogao biti u memo-polju. Kod **DBFCDX** drajvera svako memo-polje na početku sadrži dužinu, pa se u njega mogu upisivati proizvoljni simboli. U originalu, ovakva memo-polja mogu biti proizvoljne dužine, ali se javlja problem upisa i čitanja sadržaja takvih polja. Naime, *Clipper* ne podržava stringove duže od 64K, pa bi se upis i čitanje ovakvih polja morali obaviti funkcijama tipa **FRead** i **FWrite**.

Na žalost, princip alociranja memorije kod memo-polja je ostao vrlo loš: novi blokovi se upisuju na kraj, tako da je povremeno potrebno prepakovati datoteku. Naravno, ovo je potrebno raditi daleko ređe nego kod **DBT** memo polja.

Znači, **DBFCDX** je prvi konkurent za potencijalnog naslednika standardnog **DBFNTX** drajvera. Pobrojane karakteristike su velika prednost, ali ostaje još jedna bitna stvar: brzina. Ako je verovati reklami za *FoxPro*, **CDX** indeksi bi morali biti izuzetno brzi. Na žalost, iz nekog razloga **DBFCDX** drajver nije baš oličenje brzine.

Na slici 2. su prikazani rezultati nekih poređenja za drajvere **DBFNTX** i **DBFCDX**. Sasvim subjektivan utisak je da je **DBFCDX** drajver dosta razočarao po karakteristikama: da li je u pitanju loša implementacija, neodgovarajuća okolina ili nešto treće, tek „superbrzi i super mali" *FoxPro* indeksi nemaju baš zavidne performanse. Testiranje je obavljeno na 386/33MHz računaru sa 4MB RAM a konfigurisanog kao EMS. Za test su korišćene **DBF** datoteke raznih veličina, sa po 1000 – 10000 slogova. Prvi test se sastojao u tome da se izvrši indeksiranje po dva izraza: jedan je broj (N7), a drugi string (C35). Ovo je operacija koja se u praksi radi umereno često, pa brzina i nije od presudnog značaja. **DBFCDX** drajver tu ima za oko 30% lošije rezultate, što nije naročito kritično. Rezultat zavisi od strukture podataka: ako ima mnogo sličnih vrednosti indeksnih izraza, indeksiranje je značajno brže, jer se manje podataka upisuje na disk. Prikazani rezultat je srednja vrednost nekoliko merenja sa raznim podacima i pod različitim uslovima. Ono što je nesumnjiva prednost **CDX** indeksa to je veličina: kreirani **NTX** indeksi su ukupno zauzimali i po 3 puta više mesta od **CDX** indeksa sa istim sadržajem! U zavisnosti od podataka, ovaj odnos može biti još povoljniji.

Drugi test je dosta važniji: brzina rada naredbe **SEEK**, tj. traženja slučajno izabranih podataka. Ovo je najčešća operacija u praksi. I ovaj test je pokazivao razne rezultate, ali je prednost gotovo uvek bila na strani **NTX** indeksa. U mrežnom okruženju bi možda rezultati bili nešto povoljniji za **CDX** indekse zbog njihove male veličine.

Treći test je u stvari bio sledeća, veoma često ponavljana konstrukcija:

```
WHILE !Eof()
   SKIP
END DO
```

pri čemu je aktivan jedan indeks. Znači, potrebno je izvršiti prelazak na logički sledeći element koristeći aktivan indeks. Tu je vreme rada sa oba drajvera uvek bilo veoma blisko.

Poslednji test se sastoji u slučajnoj izmeni indeksa. Ova operacija se dosta često obavlja (uvek kada promenite polje koje ulazi u indeksni izraz), dobro je da bude što je moguće brža. Ovde su **CDX** indeksi prijatno iznenađenje: rezultat je sasvim neočekivano značajno bolji od **NTX** indeksa.

Sva testiranja su obavljana u raznim uslovima, sa različitim podacima, i po broju i po vrsti. Osim toga, svaki test je rađen sa i bez aktiviranog programa za keširanje diska. **CDX** indeksi su imali malu prednost jedino kod veoma malih **DBF** datoteka (oko 1000 slogova). Sve u svemu, mislim da je **CDX** drajver prilično podbacio. Zašto je to tako, teško je reći, najverovatnije se radi o propustima u realizaciji. No i pored toga, **DBFCDX** drajver ima nekoliko izuzetnih prednosti koje lako mogu biti presudne za vaš izbor, ako vam performanse nisu najvažnije.

RDD, SUTRA

RDD je svakako ideja koja ima budućnost. S obzirom da su računarske mreže sve aktuelnije, da je sve češći slučaj povezivanja raznorodnih programa, logično je očekivati da u toj sveopštoj integraciji dođe na red i

problem različitih formata podataka. Kada se zna da je firma *Computer Associates* poznata po proizvodima za velike računare, postaje jasnije zašto su uopšte kupovali *Nantucket* i *Clipper*.

Treba znati da *Clipper* poznaje osim nabrojanih **RDD**-a i drajvere sa **ograničenim** mogućnostima. Ti drajveri nemaju sve mogućnosti kao „pravi", ali mogu da služe kao filteri za ulaz/izlaz podataka u raznim formatima. Primeri takvih drajvera su **SDF** i **DELIM**, koji se koriste u naredbama kao što je **APPEND FROM** i **COPY TO**.

U vrlo bliskoj budućnosti treba očekivati pojavu novih drajvera. To je sigurno najlakši način da se premoste barijere između već uređenih programa i podataka koje ti programi treba da pristupaju. Insistiranje na novoj terminologiji u *Clipper*-u 5.2 nije slučajno i očigledno je da nas **CA** sprema za uvođenje u neke druge baze podataka.

Naročito je interesantna ideja o **RDD** u koji bi radio sa podacima u RAM memoriji: pristup tim podacima se ni po čemu ne bi razlikovao od pristupa standardnim podacima u **DBF** datotekama (znači, obavljao bi se standardnim **USE**, **REPLACE**... naredbama), ali bi rad sa njima bio ultra brz. Ne postoji nikakva prepreka da se u okviru takvog drajvera prvo ispita da li ima dovoljno memorije, pa ako ima de se podaci drže u njoj, a u suprotnom na disku. Što je najlepše, sve to je potpuno nezavisno od programa: možete sasvim mirno pisati program ne razmišljajući gde će biti smešteni podaci koji se obrađuju.

TERMINALSKI DRAJVERI

Izgleda da je ideja **RDD**-a ponela autore *Clipper*-a da izvrše generalizaciju i u drugim podsistemima, neke da postojeći, pa kasnije izbačeni drajveri za ekran ponovo su se našli u spisku datoteka. Radi se o tri drajvera **ANSITERM**, **NOVTERM** i **PCBIOS**.

Clipper-ove funkcije za ispisivanje na ekran ne koriste odgovarajuće **BIOS** funkcije, već sve upisuju direktno u video memoriju. Ovo je daleko najbrži način i bez problema radi na svim standardnim **PC** računarima. Međutim, problem nastaje kod onog „skoro svim": pojedine mašine imaju specifičan hardver, slika se na ekranu ispisuje preko raznih drajvera koji presreću **BIOS** funkcije. Tako da program pisan na *Clipper*-u tu jednostavno neće dobro raditi. Drugi jednostavan primer je redirekcija rezultata: ako napravite jednostavan *Clipper* program koji ispisuje neke podatke na ekran i ako zaželite da ih memorišete u datoteci klasičnom redirekcijom **PROG.EXE > PROG.LOG**, moraćete da menjate program, sav izlaz ide samo u video memoriju i datoteka **PROG.LOG** će ostati potpuno prazna.

ANSITERM je drajver koji pri ispisu na ekran pretpostavlja da je reč o standardnom **ANSI** terminalu koji prepoznaje tzv. *escape* sekvence. Na primer, ako na **ANSI** terminal odštampate **<Esc>[2J**, izbrisaće se ekran. Da bi vaš program koristio ovaj drajver, dovoljno je u komandu za linkovanje dodati datoteku **GT.OBJ** i biblioteku **ANSITERM.LIB**. Da biste videli rezultat na standardnom **PC** računaru, u **CONFIG.SYS** morate ubaciti direktivu

DEVICE=ANSI.SYS

Sve funkcije za rad sa ekranom će raditi sasvim uobičajeno, jedino će ispis biti daleko sporiji.

Drajver **NOVTERM** je namenjen mrežnim serverima koji zbog konflikta sa *Clipper*-ovim sistemom sporije rade sa štampačem. Drajver **PCBIOS** umesto direktnog upisa u video memoriju za ispisivanje koristi standardne **BIOS** funkcije. On može biti koristan kada je potrebno startovati *Clipper* aplikaciju sa nekog udaljenog računara: postoji dosta programa koji presreću **BIOS** pozive i usmeravaju ih na drugi računar preko modema ili neke druge veze. Naravno, i u ovom slučaju je ispis nešto sporiji.

Sve u svemu, čini mi se da su **RDD** i terminalski drajveri sasvim očigledne najave *Clipper*-e za druge operativne sisteme.

OSTALE NOVINE

Osim **RDD**-a, koji je sigurno udarni novitet, *Clipper 5.2* ima još dosta manjih novina. Jedna od takvih je i poboljšano upravljanje memorijom. Kod programa koji koriste *expanded* memoriju uvek je bila kritična kompatibilnost sa raznim drajverima. Tu *Clipper 5.01* nije pravio problema i lepo je sarađivao sa većinom drajvera. Verzija 5.2 je, kako piše u dokumentaciji, otklonila izvesne probleme u korišćenju i dodala podršku za *LIM 4.0* kompatibilne drajvere.

Prelaz sa *Summer'87* verzije na 5.01 je doneo dosta strožiju sintaksu: stari *Clipper* je u slučaju sintaksnih grešaka vrlo površno obavljao posao i poruke o greškama su ponekad bile zbunjujuće. Verzija 5.01 je donela pomak ka strožijoj sintaksi, a verzija 5.2 je nastavila tim putem. Na primer, sad nećete moći da pri otvaranju **DBF** datoteke koristite alias **12_dat**, jer ovo nije dozvoljen identifikator u *Clipper*-u (identifikator ne sme počinjati cifrom). Što se tiče rada sa bazom podataka, novi *Clipper* vam sada neće dozvoliti da jedan alias koristite pri otvaranju dve **DBF** datoteke: biće prijavljena greška ako koristite funkcije za pristup bazi podataka, a da prethodno niste otvorili ni jednu datoteku. Takođe je urađena provera argumenata kod mnogih standardnih funkcija.

Error sistem dozvoljava dobru kontrolu grešaka u programu, ali postojali su izvesni problemi: neke greške jednostavno nisu prosleđivane sistemu za obradu grešaka, pa je program u takvim slučajevima neminovno „iskakao". Radilo se o tzv. **internim** greškama. Ove greške se nisu uvek javljale u nepredvidivim okolnostima, na primer, pokušaj upisa na zaštićenu disketu rezultuje internom greškom. Ovo je vrlo neprijatno, jer se ovakve situacije redovno sreće. *Clipper 5.2*, na žalost, u ovom pogledu nije doneo značajniju promenu: *Error* sistem i dalje nema kontrolu nad internim greškama. Ali sad su bar autori *Clipper*-a javno „priznali" da ovakve greške postoje i u okviru zvanične dokumentacije objasnili zbog čega nastaju takve greške i kako ih izbeći. U nedostatku boljeg, i to je nešto.

Značajne izmene odnose se na višekorisnički rad. Prva izmena se tiče zaključavanja slogova kada više korisnika pristupa jednoj **DBF** datoteci. Do sada je bilo moguće da svaki korisnik zaključa samo jedan slog (ili čitavu datoteku) funkcijom **RLock()** i da onda menja zaključani slog. Međutim, nije bilo moguće zaključati istovremeno više slogova: svaki naredni poziv funkcije **RLock()** bi poništio dejstvo prethodne. Sada postoji funkcija **DBRLock()** koja omogućava zaključavanje više slogova u jednoj radnoj oblasti. Ova funkcija omogućava zaključavanje slogova na osnovu njihovog rednog broja tako da se zaključavanje može izvoditi bez pomeranja sa tekućeg sloga. Funkcije **DBRUnlock()** i **DBRUnlockAll()** otključavaju zaključane slogove, a **DBRLockList()** vraća niz koji sadrži brojeve zaključanih slogova.

Program **DBU** za manje izmene u **DBF** datotekama sada radi u višekorisničkom okruženju, ali to nije baš najsrećnije izvedeno. Od pozitivnih stvari: program neprestano proverava da li slika na ekranu odgovara pravoj situaciji: ako neki drugi korisnik promeni datoteku koju vi upravo razgledate, izmene će se potpuno automatski pojaviti i na vašem ekranu. Na žalost, neke operacije se sada ne mogu tek tako aktivirati: na primer, istovremena izmena više slogova zahteva da je datoteka zaključana. Za to postoji rešenje: **DBU** treba startovati sa opcijom /e, podrazumevaće se da radi u jednokorisničkom sistemu. Ipak, malo je zamorno stalno ulaziti i izlaziti iz sporog **DBU** programa zbog neke trivijalne izmene. Uzgred, *Clipper* bi hitno morao da dobije nove pomoćne programe: **DBU**, **RL**... **PE** su žestoko pregaženi vremenom!

Dibager je dobio nov izgled i nekoliko interesantnih novina, od kojih je najkorisnija komanda za izvršavanje programa do sledeće procedure. Kada aktivirate tu komandu, program se izvršava do prvog narednog poziva funkcije ili procedure.

Postoji još čitav niz manjih izmena, počevši od uklonjenih grešaka (na primer, konflikt *MS DOS*-a 5.0 i **PLL** biblioteka), pa do potpuno novih funkcija.



PROBLEMI I DILEME

Clipper verzije 5.2 je svakako mnogo zreliji i sigurniji proizvod od sada već stare verzije 5.0. To je uglavnom zato što nije urađena nijedna krupna izmena u osnovnim sistemima, već su iskorišćene neke mogućnosti pripremljene u ranijim verzijama (**RDD**). Naravno, nema idealnih stvari, pa svaki program mora imati i grešaka. Samo se postavlja pitanje koliko su one neprijatne.

Prilikom pripreme ovog teksta isprobao sam razne drajvere za pristup bazi podataka. Test se sastojao u kreiranju **NTX**, **MDX**, **CDX** indeksa za jednu **DBF** datoteku. Program za testiranje je izgledao otprilike ovako:

USE Test VIA "DBFCDX"
INDEX ON Izraz TO Test

Naravno, za svaki drajver je menjana opcija **VIA**. Drajver za **MDX** indekse je jednostavno odbio da radi, prijavivši grešku kod naredbe **USE**. Greška je bila u tome što ne postoji datoteka **TEST.MDX**. Naravno, to je sasvim logično, jer tu datoteku tek treba da kreira sledeća naredba! Drajveri **DBFNTX** i **DBFCDX** su radili normalno. Kada sam korišćenjem programa **DBU** izvršio naredbu **PACK**, sve je proradilo bez problema. Misterija (a i krivica) je u **DBFCDX** drajveru: on iz nekog razloga prilikom indeksiranja menja jedan bajt u originalnoj **DBF** datoteci. Taj bajt se ne koristi kod **NTX** indeksa, ali očigledno da ima neko značenje za *dBASE IV*. Na nesreću, u literaturi se ovo uopšte ne pominje, tako da misterija (a i sumnja) ostaje.

Ovo nije jedini problem: pojedini korisnici verzije 5.2 su uočili probleme sa upotrebom makro operatora kod otvaranja datoteka, i u radu novog sistema za obradu grešaka kod **TBrowse** objekata. Takođe, neke naredbe i funkcije ne rade baš sasvim logično, a naročito ne onako kako piše u dokumentaciji. Ukratko, problema ima, ali za sada se može reći da nisu veliki.

Clipper 5.2 ima zaista odličnih novina i poboljšanja u odnosu na verziju 5.01, tako da se prelazak na novu verziju sigurno isplati. Čini mi se da će prelazak na svaku narednu verziju *Clipper*-a zahtevati rešavanje navedenih problema, pa se taj posao svakako mora uraditi. Poseban problem su **novine** ove verzije, najviše **RDD** drajveri. Prvo, prelazak na neki drugi format različit od **DBFNTX** nije baš tako bezbolan: na primer, za prelazak na **DBFCDX** je potrebno konvertovati sve datoteke sa memo-poljima, **NTX** indekse će zameniti **IDX** indeksi. Da bi se u rad uveli i **CDX** indeksi, neophodno je menjati program, što je posao koji bi svako uradio sa zadovoljstvom kad bi kao rezultat dobio kvalitetne **CDX** indekse. Na žalost, kvalitet drajvera za **CDX** indekse ozbiljno ugrožava ovu motivaciju. Znači, nove stvari treba oprezno uvoditi u programe, jer može biti problema. Ako vam je lakše, razmišljajte na sledeći način: život bi bio tako dosadan da nije interesantnih problema!

dBASE DEVEDESETIH

Trend postepenog udaljavanja od DOS-a konačno je počeo da se oseća i na planu baza podataka – sve više softverskih firmi predstavlja pakete za obradu podataka prilagođene *Windows*-u i OS/2. Značajno mesto, kao i obično, pripada *Microsoft*-u, koji je početkom ove godine predstavio *Access* 1.0.

Dejan Ristanović

Gledajući *Clipper* programe zahvaljujući kojima su se PC računari probili u hiljade domaćih firmi, čovek bi pomislio da za grafiku nema baš mnogo mesta – jeste da bi ekrani bili lepši, ali se imena firmi, podaci o fakturama, magacinu i tome slično sasvim lepo unose i u tekstualnom modu. Ponekad se, doduše, ukaže potreba da se nekom slogu pridruži i slika, ali se to uz manje ili više problema može rešiti uz pomoć *Clipper*-a, dodatnih biblioteka za njega (npr. *Flipper*) ili, u krajnjem slučaju, C i asemblerskih potprograma. Međutim, verovanje da za grafiku nema mesta potrajaće samo dok ne vidite prvu *Windows* bazu podataka, na primer *Microsoft Access* koji ovde prikazujemo: sličice priključene uz slogove su još i najmanji dobitak, prednosti se pokazuju u vidu znatno opširnijih maski kojima se može lako manipulisati, očigledno prikazanim relacijama između podataka, graficima i dijagramima poslovanja, „pridruženoj" obradi teksta i, što je možda najvažnije, u visokom stepenu integracije sa pratećim *Windows* programima. Posle malo eksperimentisanja, učiniće vam se da su DOS aplikacije produkti „kamenog doba" i da su *Windows* baze prava stvar – za one koji su spremni ne odgovarajuću cenu!

SVE IMA SVOJU CENU

Pod cenom ne podrazumevamo toliko sam *Microsoft Access* koji je, verovatno u cilju bržeg proboja na tržište, razmerno jeftin, naročito za američke džepove. Radi se, pre svega, o investiciji u hardver. *Access* uopšte ne radi na 8086 i 80286 računarima, a ni 80386 mu nije primeren – na „klasičnoj" 386DX/33 MHz/64 K *cache* ploči njegove performanse nisu baš sjajne i korisnik mora da bude spreman na prilično čekanje. Naročito nas je iznenadilo što se, po izlasku iz programa, iscrta uobičajeno *Windows* okruženje ali bez ikona na njemu i onda PC gotovo pola minuta nešto piše po disku, verovatno ažurirajući i zatvarajući datoteke. Ukratko, nepojmljivo sporo – realan minimum za korišćenje *Access*-a je 80486 računar na 33 megaherca i sa 8 megabajta memorije – ako je procesor brži, *Access* još lepše radi. Na 80386 računarima se može lepo eksperimentisati, ali ne preporučujemo profesionalnu upotrebu.

Osim računara i programa, potreban vam je i operativni sistem, što bi svakako trebalo da bude DOS 6.0 (ili 5.0) i *Windows 3.1* – *Access* sarađuje i sa *Windows*-om 3.0, ali tada ostaje bez nekih značajnih stvari, npr. bez povezivanja sa drugim aplikacijama. Ukratko, u instalaciju ima smisla krenuti tek pošto je obezbeđen brz računar, dosta memorije i poslednja verzija operativnog sistema – tada će se sve svesti na umetanje 7 AT disketa i čitanje raznih reklamnih poruka dok se podaci kopiraju na disk. Jednom instaliran, *Access* zauzima oko 12 megabajta prostora na disku, pri čemu se ovo zauzeće može gotovo prepoloviti uklanjanjem demo-primera i manje važnih dodataka. sve u svemu, veoma puno u poređenju sa starim dobrim dBASE-om III koji je zauzimao 500 K, ali i dalje mnogo manje od nekih *Windows* aplikacija sa kojima se ne „razgovara" bez 30 megabajta na disku.

FILIGRANSKE IKONE

Startovanje programa ne donosi ništa posebno novo – statusna linija u vrhu, meniji ispod nje, lasterski meni u dva reda i ostatak ekrana posvećen podacima koji se obrađuju. Treba, ipak, primetiti da se *Microsoft* odlučio za preorijentaciju na lasterski meni – kod ranijih *Windows* programa to je bio samo niz od nekoliko najčešće korišćenih funkcija, dok se većina ostalih operacija obavljala iz „pravih" menija. Kod *Access*-a gotovo sve može (a mnogo toga i mora) da se uradi preko lasterskog menija. Ne bismo baš mogli reći da smo se tome obradovali – lasterski meni je pretrpan ikonicama čiji je smisao prestao da bude očigledan. Lako je pogoditi da lenjir i šestar označavaju dizajniranje, a dvogled pretraživanje, ali se pojavljuje čitav niz veoma sličnih vinjeta koje se sastoje od nekoliko paralelnih linija. Te linije jednom predstavljaju uravnate radove, drugi put tabelu, treći put svojstva nekog objekta – razlika je u nekom sitnom elementu koji se na prvi pogled i ne zapaža! Dodatno ćete biti hendikepirani ako radite u 800*600 modu na 14-inčnom monitoru, pošto će tada sličice biti još sitnije i prema tome još teže prepoznatljive. Smisao većine ikona je, srećom, opisan na zadnjoj korici *User's Guide*-a koji ćete prvih dana neprekidno držati pored sebe. Posle izvesnog vremena privikavanja, ikonice će biti sve očiglednije i očiglednije, ali bi poneko mogao i odustane od čitavog programa zato što mu se stalno dešava da pozove pogrešnu stvar – to nije mali gubitak vremena, naročito na sporijem računaru. Možda bi malo više stavki u menijima olakšalo korišćenje *Access*-a – avaj, smo najzad pismeni, a igri asocijacija sa *Microsoft*-ovim programerima nije svako vičan.

KLASIČNA ORGANIZACIJA

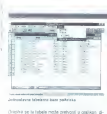
Microsoft Access je klasična relaciona baza podataka čiji su se autori opredelili za preciziranje nekih već usvojenih pojmova. Prikazujući *FoxPro*, na primer, primetili smo da se kroz dokumentaciju izrazi „baza podataka" i „tabela" koriste kao sinonimi, premda to, strogo govoreći, nije sasvim opravdano. *Access* poznaje **tabelu** koja je zapravo ono što smo u dBASE-u zvali „bazom podataka" – niz slogova jednake strukture, pri čemu se svaki slog sastoji od nekoliko polja. Jedan posao obično obuhvata veći broj tabela, i njih *Access* integriše u jednu **bazu podataka**. Govoreno dBASE-ovskim rečnikom, DBF fajl bi odgovarao tabeli, a grupa DBF fajlova koji su smešteni u isti **katalog** (jer pripadaju istom poslu) predstavlja bazu podataka. U bazu su integrisani i ekrani (formulari), izveštaji, složeni upiti, indeksi i, uopšte, sve što se u svakodnevnom radu koristi. Standardna ekstenzija *Access* baza podataka je .MDB.

Različite tabele koje čine istu bazu podataka međusobno su povezane **relacijama**. *Access* dopušta postojanje samo jedne relacije između bilo koje dve tabele i prikazuje je elektronom izlomljenom linijom. Ukoliko, na primer, u jednu tabelu upišemo podatke o kupcima, u drugu podatke o proizvodima, a u trećoj evidentiramo transakcije, svaka transakcija će na neki način biti povezana sa tabelom kupaca (da bi se videlo kome je roba isporučena) kao i sa tabelom proizvoda, da bi se znalo šta je kupljeno. *Access* obezbeđuje povezivanje relacijama 1:1 i 1:N. Relacija 1:N znači da, u datom primeru, jedan kupac potencijalno figuriše u više transakcija. Relacija 1:1, premda je *Access* podržava (uz uslov da pridružena polja moraju biti glavni ključevi u odgovarajućim tabelama), retko je u upotrebi, jer ako jedan kupac vrši tačno jednu nabavku, tabela kupaca, strogo uzevši, i nije potrebna: podaci o njima mogu da se upišu u tabelu transakcija. Relacija N:1 nije potrebna (može se posmatrati kao 1:N ali „gledana" sa suprotne strane), dok N:N relacija (slobodne veze transakcija i kupaca) nije direktno podržana, što je uobičajena slabost DBMS programa za personalne računare.
N:N relacije se, u takvim uslovima, obično realizuju kreiranjem dodatnih tabela, što može negativno uticati na performanse čitavog sistema.

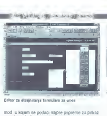
IZDVAJANJE GRUPE SLOGOVA

Sledeći važan pojam koji je *Access* „vratio u život" jeste *dynaset*: na starom dobrom BBC B i njegovoj bazi podataka *StarBase* zvao se *subset*. Ideja je u tome da se, za potrebe pregleda, štampanja, ispravija ili neke slične operacije izdvoji neka grupa slogova, otprilike kao kada u dBASE-u definišete filtar – mada svi znaju koliko filtri usporavaju rad. *Access*-ov *dynaset* je po mnogo čemu sličan pravoj tabeli – može se prikazati kao tabela, umetnuti u sve formulare i izveštaje, zumirati na čitav ekran pritiskom na Shift F2 ili, u krajnjem slučaju, pretvoriti u tabelu, ako se zaključi da je to potrebno. Međutim, *dynaset* se, po završetku operacije ili niza operacija, naprosto „zaboravlja" – podaci su sve vreme bili u svojim matičnim tabelama pa su sve eventualne promene tamo i zapisane. Vredi posebno istaći da su operacije sa *dynaset* strukturama izuzetno brze, čak i ako se oni sastoje od polja iz različitih tabela. Jedini primećeni problem je u tome što se pri ispravljanju podataka u *dynaset*-u ne konsultuju validacioni kriterijumi, što znači da provera ispravnosti podataka nije obezbeđena. Ovo smatramo ozbiljnim nedostatkom, pošto se u docnijem radu sa podacima nikada ne može sa sigurnošću računati na sadržaje nekih polja, čime čitav sistem validacije unosa postaje gotovo besprednetan.

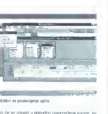
Dynaset se kreira pomoću upita (*query*), pri čemu *Access* poznaje čak tri vrste upita. Prvi od njih je *select query* koji je ujedno i najbliži onome što reč „upit" znači – zadajemo uslov prema kome se iz tabele izdvajaju neki podaci. *Action query* će, osim (opcionog) izdvajanja podataka, na njima izvršiti i neku transformaciju, recimo upisati u polje „cene" bivši sadržaj istog polja pomnožen nekom konstantom ili možda kursem marke preuzetim iz druge tabele. Izalo su „izvršni upiti" podeljeni na *make table*, *delete*, *append* i *update* akcije. *Crosstab query*, najzad, obezbeđuje neku vrstu rekapitulacije podataka iz niza tabela – uspostavljaju se relacije i podaci ispisuju u obliku tabele koja mnogo podseća na *spreadsheet* programe, a pomoću *Microsoft*



Jednostavna tabelarna baza podataka



Editor za dizajniranje formulara za unos



Editor za postavljanje upita

Graph-a se ta tabela može pretvoriti u grafikon, dijagram ili neki drugi oblik pogodan za prezentaciju. *Crosstab query* je u neku ruku proširenje standardnog *select query*-ja i često zahteva parametre, pa se kombinuje sa takozvanim parametarskim upitima (*parameter query*) koji zapravo predstavljaju prelaz sa upita na programski jezik.

PITAJTE SLOBODNO

Autori *Access*-a učinili su sve što se moglo učiniti da omoguće onima koji nemaju posebno programersko obrazovanje da zadaju čak i veoma složene upite. Tu je pre svega mogućnost zadavanja upita preko primera (QBE, *Query By Example*) gde se u masci biraju pojedini slogovi i polja, a program samostalno „shvata" šta je korisniku potrebno i izdvaja ceo *dynaset*. Takvo zadavanje je, naravno, ograničeno na jednostavne upite. Korisnici nešto većih zahteva moraju da popunjavaju svojevrsnu tabelu – vrste te tabele su zapravo zamena za operator ili (*or*), dok su kolone zamena za i (*and*). To praktično znači da za zadavanje uslova „broj između 10 i 100 ili između 200 i 300" treba rezervisati dve vrste – u prvoj piše *Between 10 And 100* a u drugoj *Between 200 And 300*. Ako se umesto brojeva zadaju neki drugi podaci, treba zapamtiti da se datumi navode između znakova # (npr. #01 Jun 1993#) i da se imena polja uokviruju srednjim zagradama (npr. [Cena]). Kod složenih uslova neka polja se moraju navoditi u više vrsta, a moraju se zadavati i neki dodatni uslovi da bi ceo upit bio konzistentan.

Pokazuje se, zapravo, da za veoma složene upite nema druge nego otkucati čitav logički izraz na način uobičajen za većinu programskih jezika i diktiran sintaksom internog SQL-a (*Structured Query Languge*). Početnici kojima su potrebni tek osnovni upiti će se ipak sa *Access*-om snaći mnogo lakše nego sa dBASE-om ili čak *Paradox*-om.

Pri zadavanju upita posebnu ulogu igraju takozvani džoker znaci – gde god se očekuje neka vrednost, može se uneti službena reč *Like* i iza nje izraz u kome figuriše upitnik, zvezdica ili „taraba". Kao što je i uobičajeno, upitnik je zamena za bilo koji znak, zvezdica za bilo koji niz znakova, a „taraba" (#) zamenjuje bilo koju cifru. Pri korišćenju džoker znakova nema ograničenja na koja nas je DOS naučio – zvezdica može da se nađe i na početku reči ili se za traženje datuma može zadati i nešto kao */*/93, što „otkriva" bilo koji ovogodišnji datum. Po potrebi se zadaju i razni izrazi: recimo [A,B] znači slovo A ili slovo B, [A-M] je slovo „između" A i M, a [!S] označava bilo koje slovo osim S.

Obezbeđene su i komande za pronalaženje praznih polja (*Is Null*) ili zahteva da traženi sadržaj bude jednak kompletnom polju, nekoj reči u njemu (reč je za *Access* niz slova oivičen blankovima), da se nalazi bilo gde u polju ili da bude sadržan u bilo kom polju datog sloga. Za razliku od dBASE-a može se zahtevati pretraživanje unapred i unazad, a vredi pomenuti i takozvani *search as formatted* mod, u kojem se podaci najpre pripreme za prikaz, a onda pretražuju – na taj način se, recimo, može tražiti reč *Feb* u sistemski formatiranom datumu.

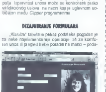
KLJUČEVI I INDEKSI

Kada se radi o indeksima, *Access* je relativno sličan mnogim ranijim programima. Svaka datoteka, pre svega, mora imati primarni indeks koji zapravo predstavlja neku vrstu zamene za redni broj sloga – ako ga sami ne kreirate, *Access* će automatski kreirati takvo polje i obezbediti unos podataka u njega – svaka sledeća vrednost biće za jedan veća od prethodne. Primarni indeks ne mora biti numeričko polje, ali to najčešće jeste – u svakom slučaju, dva sloga ne mogu imati jednak primarni indeks, pa je zahvaljujući tome pretraživanje po njemu praktično trenutno, ali se zahteva puno navođenje indeksa, a ne, recimo, prvih nekoliko njegovih znakova. Ostalim poljima po potrebi se pridružuju sekundarni indeksi kojih može biti više i koji nisu ograničeni na vrednosti polja – mogu ih zameniti i izrazi. Uveden je i pojam stranog ključa (*foreign key*) preko kojeg se realizuju relacije između različitih tabela – primarni ključ prve tabele ugrađuje se u drugu kao strani, pa se preko njega kontroliše koncept simultanog kretanja kroz relacione baze podataka.

Indeksi se održavaju potpuno automatski – kada pomeranjem kursora iza poslednjeg popunjenog reda tabele kreirate novi slog, *Access* će automatski popuniti eventualni glavni ključ i dodeliti preostalim poljima podrazumevane vrednosti, ako su one zadate prilikom formiranja baze. Pri zadavanju tih vrednosti mogu se, osim konstanti, koristiti i izrazi u kojima figurišu sadržaji već popunjenih polja. Ispravnost unosa može se kontrolisati preko validacionog uslova, na način koji je uglavnom uobičajen među *Clipper* programerima.

DIZAJNIRANJE FORMULARA

„Klasični" tabelarni prikaz podataka pogodan je za neke najelementarnije operacije, ali za komforan unos ili pregled treba poraditi na masci – podaci će se unositi u pogodno raspoređena kućice, logička polja će se „overavati" iksom u odgovarajućem kvadratiću, za izbor jedne od nekoliko dopuštenih vrednosti koristiće se meni i tome slično. Samo se po sebi razume da će formulari u *Windows* okruženju biti znatno efektniji od onih na koje nas je *Clipper* navikao – tu su trake za skrolovanje, *pull down* meniji, slike i štošta drugo.



Kombinovanje slika i teksta u istom slogu

Znajuć da pri programiranju na *Clipper*-u dobar deo vremena odlazi baš na raspoređivanje podataka po ekranu, pomislilo bi se da je pod *Windows*-om ovaj zadatak izuzetno složen i nezahvalan. Daleko od toga – zahvaljujući *Form Wizard*-u u mnogim slučajevima će ga obaviti sam *Access*!

Form Wizard-u treba samo reći koja polja treba da se jave u masci i da li da ona bude jednokolonska (jedan slog na ekranu), tabelarna, organizovana na principu formulara i podformulara ili čisto grafička, a on će racionalno rasporediti polja, dodati odgovarajuće komentare i uopšte spremiti sve za unos podataka. Kada se radi o složenijem unosu koji se zasniva na relacijama između raznih tabela, *Form Wizard* verovatno neće doveсti od finalnog rešenja, ali svakako može da uradi početni i najdosadniji deo posla – posle se pomoću posebnog editora maska može prilagoditi i usavršiti.

Rad sa editorom će mnoge podsetiti na crtanje u *PC Paintbrush*-u ili nekom sličnom programu, ali verujemo da imamo mnogo bolju analogiju – ako ste nekada na *Spectrumu* koristili *Pinball* i u okviru njega dizajnirali fliper automate, editor formulara vas neće ničim iznenaditi! U donjem levom uglu ekrana su alatke koje uzimate i raspoređujete po radnom prostoru. Osim osnovnih alatki (polje, komentar itd.) možete da uključujete i menije (*Combo box*), grupe opcija (*option group*) koje rezultiraju numeričkom vrednošću date maske i svašta drugo. Posebno su zanimljiva višelinijska tekstualna polja, pri čijem popunjavanju deluje čitav mali tekst editor – može se obezbediti da svaki od redova bude „beskonačno" dugačak, tj. da se sadržaj polja skroluje po horizontali a da se u sledeći red prelazi pritiskom na Enter, ili da čitavo polje predstavlja jedan tekst, tj. da sam program prebacuje reči u sledeći red i uopšte formatira tekst ako se neki njegov segment produži ili skrati. Može se operisati čak i sa veličinom samog polja na ekranu – dalji unos u puno polje se zabranjuje, sadržaj skroluje naviše ili se čak čitav prozor povećava na račun nekih susednih polja (*Can Grow / Can Shrink* opcije).

Primetili smo i nekoliko alatki dobro prilagođenih *Windows* okruženju. Pre svega, izgleda da je došao kraj zadavanju dužina polja u znacima – *Windows* podržava toliko raznih fontova da je termin poput „pet slova" postao previše neprecizan. Dužine se, dakle, zadaju u inčima (ili milimetrima), a *Access* samostalno vodi računa o ograničenjima navedenim pri dizajniranju baze – ako su slova krupna, deset znakova će zahtevati horizontalno skrolovanje, dok će kod nekog veoma sitnog fonta možda učiniti da polje bude „puno" iako grafički deluje poluprazno. Tu su i alatke za vertikalno i ho-

rizontalno „uravnavanje" polja (već u 800•600 rezoluciji teško je biti siguran da je jedno polje tačno ispod ili pored drugog), kao i Tool lock funkcija koja je tako nedostajala Spectrum-ovom Pinball-u. Tu su, najzad, i neke posebne funkcije, kao što je „taster" *Open Another Form* koji omogućava međusobno povezivanje različitih formulara. To će se ipak koristiti tek kod ekstremno složenih unosa – u normalnim uslovima je sasvim dovoljno što se formular može prostirati preko više ekrana, po kojima se korisnik kreće pomoću traka za skrolovanje ili pritiscima na PgUp i PgDn.

BOGATI IZVEŠTAJI

Izveštaji su, po mnogima, ono po čemu se DBMS programi porede – na kraju krajeva, izveštaj je gotovo uvek finalni proizvod, ono što šaljemo dalje i na osnovu čega se donose odluke. Zato su autori *Access*-a posvetili dosta pažnje izveštajima: kreiran je, zapravo, još jedan od njihovih „softverskih čarobnjaka" zvani *Report Wizard*.

Report Wizard, nalik na *Form* „saučesnika", zahteva da imenujete polja koja se pojavljuju u izveštaju i da izaberete tip: *single column*, *graph*, *totals* ili *mailing labels*. Zatim se izveštaj formira i, po aktiviranju odgovarajuće opcije, prikazuje na ekranu ili štampa na nekom od *Windows* štampača (npr. *LaserJet IV*). Pokazuje se, na žalost, da su obični koje kreira *Report Wizard* previše rudimentarni da bi mogli predstavljati kompletan izveštaj – u većini slučajeva to će biti samo prvi korak koji će se podešavati primenom *Report Editor*-a. On je po mnogo čemu sličan *Form Editor*-u, ali je velika pažnja posvećena radu sa fontovima i raspoređivanju grafičkih elemenata na strani – opisi, propratni tekst i sadržaji samih polja mogu se ispisati bilo kojim trenutno instaliranim fontom (uglavnom će se koristiti TTF), dok se logotip firme i druge slike automatski umeću na predviđena mesta. Izveštaj se može prostirati na više strana i u tom slučaju se može podesiti paginacija, dodati zaglavlje ili potpis, proveriti mesto na kojem se strane seku i svašta drugo.

ACCESS I OLE

Pominjanjem logotipova i drugih grafičkih elemenata došli smo do noviteta koji će svako najpre pomenuti kada se govori o *Windows* bazama podataka – integracija teksta i slika ili, u opštijem slučaju, OLE (*Object Linking and Embedding*). *Access* se i na ovom planu ponaša kao moderna *Windows* aplikacija – u formulara i polja baze podataka može se ugraditi bilo koji objekt kreiran u nekom drugom *Windows* programu: najčešće će to biti slika, ali nema razloga da se ne uključi i tabela, grafikon, tekst ili nešto treće. Pokazuje se, ipak, da sve OLE tehnike nisu primenjene – svi objekti mogu da se ugrade u bazu (*embedding bound object*), ali je povezivanje (*linking unbound object*) moguće samo sa nekim programima. U čemu je razlika? Ugrađeni objekti su deo baza podataka i njima se ne može operisati pomoću drugih programa, dok je povezani objekat samo „pridružen", tj. može ga menjati *Access* ali i program u kome je nastao. *Access*, što je i prirodno, odlično sarađuje sa *Microsoft Graph*-om (u cenu je uračunat i ovaj paket), što garantuje kvalitetnu prezentaciju obrađenih rezultata.

Pri umetanju objekata u tabele i formulare koristi se mehanizam sasvim sličan *Ventura*-nom. Osnovni pojam je ram (*frame*) u koji se objekat smešta. Način smeštanja, naravno, nije jedinstven – objekat se može dimenzionisati tako da odgovara ramu, pri čemu će njegove proporcije biti očuvane (*zoom*) ili deformisane (*scale*), a može se naložiti i smeštanje onog dela objekta koji može da stane u ram uz „odsecanje" ostatka (*clip*). Najčešće će, naravno, ram biti toliki da nedeformisani objekat tačno stane u njega.

LIČNA KARTA

Program *Access*

Verzija 1.0, 22.12.1992.

Namena Baza podataka

Sadržaj paketa. 8 1.2 M disketa, 5 knjiga i drugi materijali

Hardversko-softversko okruženje 80386 ili bolji, 4 M RAM-a, hard disk, DOS 3.30 ili noviji, Windows 3.0 ili noviji

Zauzet prostor na disku oko 12 megabajta

Dokumentacija
Getting Started (180 A5)
User's Guide (720 A5)
Introduction to Programming (168 A5)
Language Reference (550 A5)
Microsoft Graph (120 A5)

Proizvođač
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399 USA

Za kreiranje sličnih izveštaja često će se koristiti i makro naredbe koje zapravo predstavljaju neku „sredinu" između interaktivnog rada i programiranja. Makro se, kao i u mnogim drugim *Windows* programima, zadaje izborom odgovarajuće opcije i menija, i izvršavanjem niza akcija koje ubuduće treba ponavljati. To mogu biti sasvim jednostavne stvari, kao što je oglašavanje zvučnika ili kopiranje objekta, ali i vrlo složeni zadaci koji obuhvataju kreiranje i gašenje tabela i mnoge slične operacije. Makroi se mogu grupisati u skupine, ugrađivati u glavni meni ili dodeliti tasterima. Značajna je i mogućnost pozivanja makroa pri startovanju *Access*-a, što retko koja *Windows* aplikacija obezbeđuje: *Access*-om se, naime, može upravljati i iz „komandne linije", tj. iz PIF fajla kojim se program startuje. U jednostavnijem slučaju, može se po startovanju automatski otvoriti proizvoljna baza podataka prostim navođenjem njenog imena.

ACCESS ZA PROGRAMERE

Ambiciozniji korisnici svakako će poželeti da iskoriste sve mogućnosti *Microsoft Access*-a i tako se „bacit" na programiranje. Priloženi programski jezik zove se *Access Basic*, pri čemu drugu reč imena ne zaslužuje – jedva da je sličan „pravom" bejziku, a i daleko je od nekakve „elementarnosti". Radi se, zapravo, o moćnom objektnom jeziku posebno prilagođenom radu sa bazama podataka. Kada kažemo „objektni", ne mislimo baš na neki *SmallTalk* klon – korisniku nije omogućeno ozbiljno kreiranje sopstvenih klasa, ali je proizvođač predvideo razne sistemske objekte koji se docnije obilato koriste. Osnovni sistemski objekti su ekran (*screen*), alati (*tools*), formulari (*forms*), izveštaji (*reports*), tabele (*table*) i slogovi (*recordsets*). Svaki od objekata ima neka svojstva (*properties*), a njima se upravlja naredbama, metodima, procedurama i funkcijama. Ponekad se dogodi da isto ime bude iskorišćeno i kao svojstvo i kao metod, ali to pravi probleme pre svega početnicima – iskusniji korisnici na ovaj način pamte manje mnemonika.

Objektna organizacija je neke stvari pojednostavila, a neke baš i nije – imamo utisak da su se pojedine operacije nekada brže i lakše obavljale. Pomenimo, na primer, standardnu situaciju kada radimo na jednom slogu i onda treba privremeno da pređemo na neki drugi da bismo se, kada tamo sredimo ono što treba srediti, vratili polaznom podatku. U *Clipper*-u bi se to obavilo jednostavnim NN =RECNO() i docnijim GOTO NN – ne može biti jasnije, a znamo i šta NN čuva, redni broj sloga. Kod *Access*-a, stvar se svodi na *BookMark* svojstva, koja bi trebalo da predstavljaju analogiju sa ceduljicama koje umećemo u knjigu da bismo obeležili neko važno mesto ili da bismo znali gde smo prekinuli čitanje – posle **pozic**=podaci.bookmark u string *pozic* se upisuje neki tajanstveni niz slova i brojeva koji jednoznačno obeležava tekući slog. Kada želimo da se tom slogu vratimo, izvršićemo **podaci.bookmark**=pozic – nova je samo sintaksa i činjenica da više ne znamo šta kontrolna promenljiva krije! Možda će ceo rad sa objektima postati jasniji kada pogledamo jednostavan primer programa koji dodaje novi slog aktivnoj tabeli, popunjava ga vrednostima, snima na disk, pronalazi i briše.

```
Dim Baza As Database, Tabela As Table
Set Baza=CurentDB()
Set Tabela=Baza.OpenTable("knjige")
Tabela.Index="PrimaryKey"
Tabela.AddNew
Tabela("ID knjige")="12003"
Tabela("Autor")="Asimov Isaac"
Tabela("Naslov")="Zadužbine i Zemlja"
Tabela.Update
Tabela.Seek "=", 12003
Tabela.Delete
Tabela.Close
```

Izuzetno značajna mogućnost *Access*-a je rad sa transakcijama. Korisnicima *BTrieve*-a transakcije su dobro poznate – proglasimo početak transakcije i onda čitav niz radnji koje završavamo naredbom da se rezultati snime na disk ili potpuno ignorišu. Prednosti su više nego jasne – u dobro pisanim *Clipper* programima događa se da trećina koda ode na tretiranje nekih „nenormalnih" situacija koje ne bi trebalo nikada da se dese, ali bi teško oštetile podatke ako bi se desile. Kod *Access*-a sve ide automatski: transakciju ćemo komandom **Commit** finalizovati tek ako je sve dobro prošlo, dok će u protivnom komanda **RollBack** vratiti sve podatke u stanje pre **Begin Trans**. U okviru transakcija mogu se pokretati nove transakcije, i tako u pet nivoa.

Prostor u ovom prikazu ne dopušta nam da detaljnije opisujemo *Access Basic* i njegove naredbe – reći ćemo samo da je podržano sve što je potrebno da bi se, bez previše truda, napravila prava pravcata *Windows* aplikacija čije je izvršavanje, na žalost, uslovljeno postojanjem samog *Access*-a na ciljnom računaru. Na raspolaganju su, dakle, klasični specijalizovani i lasterski meniji, vinjete, *help* pozivi, veze sa drugim aplikacijama... Programiranje, na žalost, zahteva prilično navikavanje – ako ste već programirali u *Windows* okruženju, ne bi trebalo da imate probleme, ali će početnik provesti dosta vremena nad knjigom pokušavajući da razaberu šta su objekti, šta događaj (*event*), OLE... Prilikom tog pretraživanja naići će i na ponešto što se ne očekuje od DBMS programa: heksalno-oktalne konverzije, mogućnost promene donjeg indeksa već aktiviranog niza, pseudoslučajne brojeve, sve

logička operacije koje obuhvataju i implikaciju, ekvivalenciju.

Što se performansi tiče, Access nije u rangu sa dBASE-olikim jezicima na koje smo navikli – iz gleda da su autori smatrali da je brzina pretraživanja manje važna od fleksibilnosti i atraktivnosti ekrana. Jedino se pretraživanja po indeksu obavlja praktično trenutno. Primećuje se, najzad, i izvesna nedorađenost koja u praktičnom radu ume da zasmeta: pozivanje funkcije ASC sa praznim stringom kao argumentom izaziva, na primer, izvršnu grešku i prekid programa, odnosno nalaže programera da definiše proceduru za obradu grešaka.

UVOZNO-IZVOZNI POSLOVI

Kada se radi o sigurnosti, Access omogućava kreiranje korisnika i preciznu kontrolu njihovih prava. Svi koji pristupaju sistemu podeljeni su u tri grupe: administratori, korisnici i posetioci. Administrator u principu može da radi bilo šta, pa i da dodeljuje prava sebi i drugima. Korisnik ima ona prava koja mu je administrator dodelio, dok gost može samo čitati podatke ili mu se čak zabranjuje bilo kakav postup. Da bi se sistem osigurao, treba aktivirati Security meni, napraviti jednog korisnika u grupi administratora i najzad ukinuti generičkog korisnika Admin. Od tog momenta svako startovanje Access-a podrazumeva unos korisničkog imena i lozinke i pristup samo nekim bazama podataka ili tabelama. Pokazuje se, na žalost, da je sva ova zaštita površinska: neovlašćenom korisniku Access zabranjuje neke operacije, ali ga ne može sprečiti da ode u DOS i odatle obriše ili čak izmeni podatke, primenjujući neki drugi program. Najviše što administrator može učiniti je da podatke, kada nisu u potrebi, kriptografiše – radoznalci će takve podatke u krajnjem slučaju možda i obrisati, ali bar ih neće moći smisleno menjati! Ozbiljan softver za kriptografisanje se, zbog restriktivnih mera američkog Ministarstva odbrane, ne može izvoziti, pa je za evropsko tržište zamenjen jednostavnijim i mnogo slabijim.

Access programi rade pod mrežom – podržano je nekoliko mrežnih arhitektura, od kojih je za nas najznačajniji Novell. Detekcija mreže je potpuno automatska, a o zaključavanju baza podataka i slogova ne treba posebno brinuti – sve će se, bar u normalnim situacijama, izvršiti potpuno automatski. Ova automatika, na žalost, baš i ne pomaže programerima koji bi želeli da drže „sve" u svojim rukama – projektovanje aplikacija koje bi radile brže zato što „znaju" da u nekoj prilici nema potrebe zaključavati određenu datoteku (jer joj se po prirodi problema ne može pristupiti) sudariće se sa željom Microsoft-ovih programera da se što više toga automatizuje. Reklo bi se da će se ovaj sukob interesa završiti u Microsoft-ovu „korist", osim ako je korisnik spreman da piše potprograme u drugim jezicima.

Access dobro „sarađuje" sa drugim bazama podataka – mogu se preuzeti datoteke u dBASE (III i IV), Paradox, BTrieve i Microsoft SQL server formatu. Svaka od „stranih" datoteka se prevodi u jednu tabelu i uključuje u Access bazu podataka, bilo putem konverzije ili kao „pridruženi član". Konvertovane datoteke se transformišu u Access format i od tog trenutka im se pristupa znatno brže, ali su zato nedostupni programima u kojima su nastali. „Uvežene" datoteke ostaju u originalnom formatu – njihov sadržaj se ubuduće može menjati kako iz Access-a (uz određeni gubitak u performansama) tako i iz dBASE-a, Paradox-a ili uopšte programa u kojem su i nastale. Ova fleksibilnost omogućava visok stepen integracije svih podataka koji se trenutno nalaze na vašem disku: pri radu u mrežnim uslovima svaki korisnik i dalje može ostati pri programu koji mu najviše odgovara ili na koji je navikao, što ne sprečava administratore i vođu projekta da podatke dalje analiziraju uz pomoć Microsoft Access-a.

ZA I PROTIV

Hvalimo

1. Atraktivan korisnički interfejs
2. Moćan objektno-orijentisan programski jezik
3. Jednostavno i automatizovano kreiranje formulara i izveštaja
4. Visoku kompatibilnost sa drugim proizvodima
5. Izuzetnu dokumentaciju

Kritikujemo

1. Prenaglašene i nedovoljno prepoznatljive ikone
2. Sitne propuste koji otežavaju programiranje

Testirajući čitav sistem, primetili smo značajna usporenja, naročito kada se radi o dBASE fajlovima koji su nam najviše i trebali – Access je često bio dva ili tri puta sporiji od dBASE-a, naravno kada je radio sa datotekama u dBASE formatu. Posle dužeg eksperimentisanja zaključili smo da je stvar u podršci mreži i višekorisničkom radu, koja očito usporava mnoge operacije – kada smo datoteku proglasili nedeljivom (sharing disabled) rad je značajno ubrzan i uglavnom se obavljao brzinom na koju nas je dBASE navikao; performanse Clipper-a i naročito FoxPro-a nisu dostignute.

Zanimalo nas je kako se Access „snalazi" sa specifičnim oblicima zaštite koje ugrađuju autori konkurentskih baza podataka. Zato smo pomoću Paradox-a kreirali jednu bazu podataka, zaštitili je lozinkom i pokušali da „uvezemo" podatke u Access tabelu. Sve je radilo savršeno: pojavila se maska koja zahteva lozinku i, kada je lozinka uneseна, konverzija je uspešno obavljena; dobijenu tabelu je prirodno trebalo ponovo zaštititi korišćenjem samog Access-a. Tretman „pridruženih" datoteka nas je začudio – Access je prvog puta zahtevao lozinku, a docnije čak i po isključenju i ponovnom uključenju računara pristupao podacima bez potrebe da se ona ponavlja! Podaci su, ipak, i dalje bili zaštićeni, jer im se iz samog Paradox-a moglo pristupiti tek posle kucanja lozinke. Listanje dokumentacije otkrilo je u čemu je stvar: Access lozinku zapisuje u skriveni deo svoje konverzione tabele, što znači da je sigurnost čitavog sistema svesno žrtvovana na račun komfora pri docnijem korišćenju. U svakom slučaju, ne savetujemo da lozinkom zaštićene baze podataka pridružujete Access tabelama.

LIMITI ACCESS-A

Svaki program koji operiše sa podacima zasniva se na nekim limitima (dužina polja, broj polja u bazi, broj slogova, dužina datoteka itd) a mora se reći da su autori Access-a te limite malo čudno izabrali. Negde su pre svega bili više nego velikodušni: bazu podataka može da čini 32768 tabela, a istovremeno se može otvoriti 256 (oba cifra su očito predimenzionisane); svaka tabela se sastoji od najviše 255 polja (što ume da bude neprijatno malo), dok je ime svakog od tih polja ograničeno na 64 znaka, pri čemu se u imenu slobodno kombinuju slova, brojevi i specijalni znaci koji **uključuju** i blanko simbol. Ovo je, reklo bi se, veliki napredak u odnosu na dBASE-ovske „spartanske" imena, ali ipak vredi primetiti da će se retko ko odlučiti da polju da ime duže od desetak znakova. Neprijatna su i ograničenja sadržaja slogova – tekstualno polje je ograničeno na 255 znakova, a memo-polje na 32 kilobajta, što je slabije od većine dBASE klonova – FoxPro, na primer, ograničava memo-polja jedino prostorom na disku. Sama baza podataka, koja uključuje svih 32000 tabela, ograničena je na 128 megabajta. TO je prilično veliki broj koji ipak ne mora biti dovoljan – u Jugoslaviji, na primer, trenutno ima oko 120.000 preduzeća, što znači da bi kilobajt po svakom od njih gotovo dosegnuo limit. Možete se snalaziti povezujući razne baze i tako na disk smestiti onoliko podataka koliko imate slobodnog mesta, ali je ipak čudno da su se autori Access-a opredelili za ovako odsečan limit. Ne baš se srećom nisu rešili kada su dimenzionisali numerička polja – brojevi se upisuju u 1, 2, 4 ili 8-bajtne promenljive, što obezbeđuje preciznu evidenciju brojeva čija je apsolutna vrednost manja od 922 337 203 685 477. Uzimajući mesečnu inflaciju od 100 posto, čak i za 20 meseci ćemo u numeričko polje moći da upišemo iznos od 1000 maraka izražen u dinarima (dvadeset i prvog meseca ćemo morati ozbiljno da porazmislimo). Ostala ograničenja su manje više očekivana: 32768 upita po bazi, 255 aktivnih korisnika, 32 indeksa, 999 akcija u makrou, izveštaji do formata 55×508 centimetara.

Dokumentacija koju Microsoft isporučuje zaslužuje najvišu ocenu. Tu je pre svega izvanredni User's Guide koji, ne podrazumevajući gotovo nikakvo predznanje, uvodi korisnika u svet baza podataka i objašnjava mu sve, od samog pojma tabele do kreiranja složenih maski, izveštaja i drugih elemenata. Što je najvažnije, sve je propraćeno efektnim primerima zasnovanim na demonstracionim bazama podataka koje ste dobili uz Access. Kada savladate ovu knjigu, možete da pređete na Introduction to Programming i tako „pređete" osnove Access Basic-a. Za pravo programiranje moraćete da listate Language Reference – tamo se kriju podaci o klasama, svojstvima, metodama, procedurama, funkcijama, sve opet prošarano sa dosta primera. Ostale knjige i knjižice ređe ćete prelistavati.

Kada se radi o bazama podataka, PC računare su proslavili dBASE i Clipper. Prvi je izuzetno zgodan za kreiranje i interaktivno manipulisanje bazama podataka koje u suštini rešava za svoje potrebe. Clipper je programski jezik koji omogućava razvoj programa za tržište. Naš je utisak da je Microsoft Access zapravo dBASE sledeće generacije: izuzetno je atraktivan, efektan, komforan, brz (uz pravu mašinu) i uopšte omogućava svima koji imaju potrebe za tim da lako kreiraju i održavaju čak i veoma složene baze podataka. Ali, to nije alatka koja bi zamenila Clipper – nedostaje „pravi" kompajler koji generiše EXE datoteke i niz usluga koje bi kompajler opšte namene trebalo da pruži. dBASE-u nova generacije potreban je, rekli bismo, i Clipper nove generacije – sačekajmo da nam ga Microsoft (ili neko drugi) ponudi! U međuvremenu, ako želite da obrađujete baze podataka pod Windows-om, ne možete pogrešiti opredeljujući se za Microsoft Access.

Korisna adresa:

MISOFT
11000 Beograd, Skadarska 45, Tel. 343-046



MUDRAC IZ IZRAELA

Stvaranja programa za knjigovodstvo, obračun plata, vodenje magacina... dakle, svega onoga što u osnovi ima neku bazu podataka, predstavlja pravi hit među programerima. Izbor alata za razvoj ovakvih aplikacija je zaista širok. Pomenimo samo *dBase*, *FoxBase*, *Clipper*, *FoxPro*, *Clarion*, *Paradox*. U trku se od skora uključio i *Wizdom Plus*, programski paket uz čije ime stoji natpis "objektni generator aplikacija".

Pavle Peković

Wizdom nam je stigao u lepo dizajniranoj kutiji u kojoj smo pronašli dve knjige, jednu 5.25 inčnu disketu i *hard lock* koji se priključuje na paralelni port računara. Na stranicama našeg lista je nekoliko puta ukazivano na probleme sa ovim nepopularnim napravama. Iako sa vrama lasta nismo imali nikakvih probleme, moram uputiti zamerku autorima zbog ovakve vrste zaštite.

Wizdom-u je za rad potreban XT ili AT računar sa 512K osnovne memorije, DOS 3.0 ili novija verzija, i nešto više od megabajta na tvrdom disku, što su zaista skromni hardverski zahtevi. Instalacija je jednostavna i svodi se na kreiranje odgovarajućeg direktorijuma i raspakivanje fajlova, što *install* program obavlja veoma brzo.

Osnovna koncepcija *Wizdom*-a je usmerena ka što bržem i jednostavnijem kreiranju aplikacije. To je ostvareno interaktivnim vizuelnim kreiranjem svih ekranskih formi, ulaznih maski, baza i izveštaja, ali i relacija među njima. Za sve ostalo tu je *Wizdom Pascal* varijanta *Pascal* programskog jezika prilagođena ostalim delovima paketa. Sve ove forme autori nazivaju objektima koji predstavljaju osnovni element *Wizdom* aplikacije. Objekti su podeljeni u tri osnovne grupe: *Screen*, *Raport* i *Program*.

SCREEN OBJEKTI

Prvu grupu objekata sačinjavaju ekranske forme kao što su ulazne maske, labele i drugo. Svaki objekat ima svoje zaglavlje (*header*) u kojem se definišu atributi tog objekta, kao i veze sa drugim objektima. Osnovni podatak o svakom objektu je, pored imena, i tip objekta koji može biti *Text*, *Menu*, *Info*, *Table* i *Complex*, u zavisnosti da li se radi o ulaznoj masci, helpu, meniju, tabeli ili pak nekoj složenijoj formi.

Svakako su najzanimljiviji *Info* objekti kojim se definišu polja u bazi, kreira ili menja sama baza, te definiše izgled i konfiguracija ulazne maske za unos podataka. Ova tri procesa korisnik obavlja paralelno, kroz definisanje izgleda ulazne maske. To se postiže jednostavnim editorom, pomoću kojeg korisnik iscrtava izgled ekrana za unos podataka, s tim što su mu na raspolaganju i mogućnosti crtanja okvira, izbora boje, postavljanja senki i još nekoliko korisnih funkcija, od kojih je svakako najznačajnija za kreiranje novog polja u bazi. Ceo postupak je veoma sličan onom koji se koristi u *form editor*-u *dBase*-a. *Wizdom* nudi veliki izbor tipova polja: logička, string, datumska, numerička (*byte*, *int*, *longint* i *real*), *call* (meni stavke) i dr. Interesantno je da postoji i *Hebrew* (hebrejski) tip, za ispis teksta s desna na levo. To je posledica činjenice da su autori *Wizdom*-a Izraelci. Na žalost, ni traga od polja za čuvanje kraćih tekstova, pa samim tim ni editora. Pored određivanja tipa, svako polje može imati kratak help koji će automatski biti prikazan na dnu ekrana u toku rada sa tim poljem. Takođe, polje određuje i *read/write* atribut za kontrolu upisa i ispisa.

Svaka baza mora imati jedinstven glavni ključ, koji može biti sastavljen od više polja. Osim primarnih, ostavljena je mogućnost definisanja i alternativnih ključeva. Bilo koje polje se može vezati sa poljem odgovarajućeg tipa iz druge baze, uz opciju kontrole ispravnosti polja kao i eventualnog prenosa sadržaja polja iz druge baze, što se ostvaruje preko *pop-up* prozora sa tabelarnim prikazom sadržaja druge baze. Na žalost, ovde smo primetili nekoliko bagova i propusta. Na primer, *Wizdom* ne vodi računa da li zbir dužina polja prelazi 80 karaktera. Takođe, ima problema i sa postavljanjem više relacija između dve baze.

Table objekat se definiše na sličan način kao *Info* objekat. Kao što mu samo ime govori, koristi se za tabelaran prikaz podataka iz baze. *Table* objekat može biti vezan za novu ali i za postojeću bazu. To znači da sadržaj baze možemo prikazati preko *Info* ili *Table* objekta, u zavisnosti šta je u tom trenutku pogodnije. Tabela se kreira definicijom jednog reda labele kao i prostora za prikaz tabale, nazvanog *scrolling area*. Za ostalo će se pobrinuti *Wizdom*. Funkcionalno nema razlike između *Info* i *Table* objekata. Prvi se najčešće koristi za unos podataka, dok je tabelarni prikaz pogodniji za pregled i pretraživanje po bazi.

U programu je veoma korisno imati i help, a za to su upravo namenjeni *Info* objekti. U stvari, oni nisu ništa drugo do objekti za pregled ASCII fajlova sa osnovnim funkcijama za kretanje kroz fajl.

Najzad, za kompletiranje korisničkog interfejsa *Wizdom* nudi mogućnost kreiranja menija. Za to se koristi *Menu* objekat, u kojem korisnik definiše izgled menija kao i meni opcije. Kao i ostali ekranski objekti, i meniji se definišu preko pomenutog editora i to definisanjem polja koje će sadržati korisnikov meni izbor, te uokvirivanjem naziva meni stavki. Pošto *Wizdom* podržava roditelj-dete hijerarhiju objekata, vrlo je jednostavno kreirati podmenije tako što se svaki podmeni definiše kao odvojen objekat i poveže odgovarajućom relacijom sa roditeljskim meni objektom.

...I VEZE MEĐU NJIMA

Uopšte, u celom paketu je puno pažnje posvećeno vezama između objekata. Svakom od njih je moguće pridružiti više drugih objekata koji mogu biti startovani pre tekućeg objekta, po njegovom završetku, u slučaju snimanja sadržaja ali i kod brisanja. Drugi objekti se mogu startovati i preko zadatih *hot-key* kombinacija. Da bi priča bila kompletna, to isto je moguće ostvariti i iz bilo kog polja neke baze, čime se na jednostavan način realizuje provera ispravnosti unete vrednosti, ispis helpa za svako polje pojedinačno, pozivi kalendara, kalkulatora i još mnogo toga.

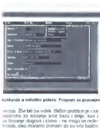
Veze između više objekata se ostvaruju preko tzv. *Complex screen* objekata. Oni rešavaju problem više različitih tipova objekata koji čine neraskidivu celinu. Primera ima mnogo. Pomenimo recimo račune čije je zaglavlje potrebno definisati kao *Info*, a ostali deo kao *Table* objekat. *Complex* objektom objedinjujemo prethodna dva i omogućujemo istovremeni rad i sa zaglavljem (*Info*) i sa stavkama (*Table*) u računu.

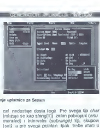
U toku izvršavanja programa, konkretno kada se radi sa *Screen* objektima, *Wizdom* korisniku pruža mogućnost da u svakom trenutku pozove meni u kojem se nalazi nekoliko korisnih funkcija i setovanja. Tu su, pre svega, opcije za izvoz podataka u ASCII, *dBase* i *Lotus 1-2-3* format, kao i pakovanje i reindeksiranje baze, zatim izbor ključa za sortiranje, kao i opcija za definisanje upita (*query*) i izlazak u DOS. Najzanimljiviji su svakako upiti, koji rešavaju pitanje pretraživanja baze. Definisanjem upita zapravo se postavlja filter i prikazuju se samo ona polja koja zadovoljavaju postavljene uslove. Uslovi se mogu postaviti za svako polje, s tim što se za jedno polje može postaviti i više uslova. Za numerička i datumska polja omogućeno je i definisanje granica, dok uslovi za string polja mogu sadržati i džoker znake * i ?. Zaista lepo zamišljeno i realizovano, ali u svemu ovome postoji nekoliko problema. Prvo, ne postoji način da se onemogući upotreba ove opcije. Drugo, mnogo bitnije, je što su sve poruke ispisane na engleskom jeziku i nemoguće ih je prevesti na naš jezik. Ovo će zaista predstavljao problem mnogim korisnicima. Zastupnici *Wizdom*-a obećavaju srpsku

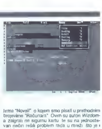
Primer 1

```
{Priprema uplatnice za stampu}
Program WIZ00014(NOVI);
SELECT UPLAT, CENOVNIK;
Begin
    getcurrent(NOVI);   { tekuci slog }
    UPLAT imepre :=NOVI imeprez;
    UPLAT adresa :=tailtrim(NOVI mesto)+    +tailtrim(NOVI ulb)
    UPLAT userid :=NOVI userid
    UPLAT username :=NOVI username
    IF NOVI new THEN
        UPLAT regnew:= NEW
    ELSE
        UPLAT regnew := ADD
    opendb(CENOVNIK);
    UPLAT cena := Cena
    PROCESS CENOVNIK    { kroz celu bazu CENOVNIK }
        UPLAT cena :=tailtrim(UPLAT cena)+    +tailtrim(str(CENOVNIK imee)),
    closedb(CENOVNIK);
    submit(UPLAT); { slanje u listu za stampu
End
```

Primer *Wizdom Pascal* programa (priprema sloga za štampu)







Aplikacija u nekoliko poteza: Program za popunjavanje uplatnice za Sezam

verziju. Živi bili pa videli. Sličan problem je i sa tasterima za kretanje kroz bazu i polja, kao i za brisanje slogova i slično – ne mogu se redefinisati, iako moramo primetiti da su vrlo logični i dosledno se koriste kroz sve oblike komunikacije sa korisnikom.

IZVEŠTAJI

Kreiranje izveštaja (*reports*) se obavlja preko *report* objekata. Svaki *report* objekat mora biti vezan za neku bazu, na osnovu čijih podataka će se generisati izveštaji. Za oblikovanje izveštaja se koristi princip vrlo sličan onom za kreiranje ekranskih formi. Dakle, potrebno je uneti željeni tekst, okvire, linije, ali i polja baze. Naravno, i ovde je moguće uspostaviti relacije između baza, te kao element izveštaja koristiti polja i iz drugih baza. Dozvoljeno je i korišćenje tzv. lokalnih varijabli za smeštanje sadržaja bitnih samo u trenutku štampe.

Svaki izveštaj se sastoji iz nekoliko celina (grupa). Prva je tzv. *Title* grupa, u kojoj se definiše ono što će se ispisati samo jednom i to na početku dokumenta. Slična je grupa *Summary*, čiji se sadržaj štampa na kraju dokumenta. Za početak i kraj svake strane postoje *Heading* i *Footing* grupe, a ostatak se zove *Detail* grupa. U okviru svake grupe se mogu postavljati i uslovi za štampu (*Condition*), kao i sistemske promenljive, kao što su tekući datum i vreme, broj strane, broj reda i tekuće linije. Podržano je i nekoliko vrsta ispisa: *bold*, *italic*, *underline* i *wide*, i to preko *Epson Escape* sekvenci. Ukoliko štampač nije *Epson* kompatibilan, potrebno je uneti odgovarajuće kodove u konfiguracionu datoteku. Ceo sistem za kreiranje izveštaja je vrlo efikasan, osim u slučajevima kada je potrebno uklopiti izveštaj u već kreirane formulare, jer *Wizdom* generator izveštaja nema opcije za fino podešavanje vertikalnog pomeranja papira. Ovo se istina može izvesti žrtvovanjem jednog od tipova ispisa, recimo *wide*, i definisanjem, u konfiguracionoj datoteci za štampač, sekvence za vertikalno pomeranje papira na mestu za *wide* kodove.

Svaki izveštaj se može poslati na ekran, štampač ili fajl, što se ili unapred definiše ili ostavlja korisniku da pred samo štampanje izabere. Takođe, definisanjem filtera, umesto štampanja cele baze, štampaju se samo željeni slogovi.

Report objekti su u hijerarhiji objekata potpuno ravnopravni sa ostalim. Dakle, moguće ih je pozivati iz *screen* i *program* objekata, ali i obratno.

PROGRAM OBJEKTI

Za kreiranje i najjednostavnijih programa neophodno je korišćenje *Wizdom Pascal*-a. Reč je o *Pascal*-u proširenom prvenstveno funkcijama i procedurama za podršku rada sa bazama podataka. U odnosu na standardni *Pascal* nedostaje dosta toga. Pre svega tip *char* (realizuje se kao string[1]), zatim pobrojani (*enumerated*) i intervalni (*subrange*) tip, skupovi (*set*) a pre svega pointeri. Ipak, treba imati u vidu da je ovaj *Pascal* deo paketa za razvoj aplikacija za rad sa bazama podataka, pa ćemo u tom smislu istaći postojanje brojnih funkcija za rad sa bazama.

Za vezu sa *screen* i *report* objektima *Wizdom Pascal* koristi tip *Object*, koji je najsličniji izvestajima. Elementu objekta se pristupa preko imena tog objekta i imena elementa razdvojenih tačkom, što se jasnije vidi na primeru 1. Boljoj komunikaciji sa ostalim objektima doprinose i funkcije za kontrolu polja, delumski i *time* tipovi, kao i nekoliko specifičnih sistemskih varijabli. Ostatak *Wizdom Pascal*-a je manje-više sve ono što smo navikli da vidimo u prosečnim implementacijama *Pascal*-a.

Wizdom editor za kreiranje *Program* objekata je više nego skroman. Na raspolaganju su osnovne blok operacije, jednostavan makro-rekorder i još nekoliko sitnica. Nema čak ni opcije za pretraživanje. Kreiranje *Wizdom Pascal* programa otežava i činjenica da ne postoji *debugger* niti išta slično tome. Izgleda da su autori *Wizdom*-a odlučili da u centar stave generatore ekrana i izveštaja, dok je *Pascal* namenjen za pomoć pri ažuriranju baza i realizovanju složenijih tokova podataka.

Svi objekti, pa samim tim i programi, se čuvaju u *Wizdom* fajlovima nedokumentovanog formata. Ovo je nezgodno iz više razloga. Prvi je svakako to da je nemoguće koristiti eksterni editor za kreiranje *Program* objekata, što bi bilo veoma korisno s obzirom da je interni editor loš. Verovatno je takva organizacija i jedan od uzroka velike sporosti pri snimanju objekata. Ovaj test smo radili na jednoj 386/25 mašini sa 8M RAM-a i diskom sa propusnom moći od preko jednog megabajta – na takvoj konfiguraciji za snimanje ekranskih objekata je nekad bilo potrebno i preko pola minuta, dok je snimanje program objekata od po dvadesetak redova trajalo i od 10 sekundi. Kompajlirani *Pascal* programi se prevode u nekakav p-kod, koji *run-time* modul tokom izvršavanja programa interpretira. Jasno je, dakle, da *Wizdom* ne pravi ništa ni nalik standardnim OBJ fajlovima, već koristi svoju organizaciju upisivanja dobijenog p-kod-a. Zbog ove zatvorenosti *Wizdom*-a zaboravite vezu sa drugim jezicima, grafiku, komunikacije i sve ono što izlazi iz okvira unosa, ažuriranja i jednostavnije obrade podataka. Ipak, moramo reći da se većina aplikacija za pomoć u poslovanju sastoji upravo od takvih operacija, te da je *Wizdom* na tom polju i te kako u igri.

UZ MALU POMOĆ BTRIEVE

Kompletan rad sa *Wizard* bazama podataka se obavlja preko *Btrieve*-a, rekord-menadžera firme "Novell" o kojem smo pisali u prethodnim brojevima "Računara". Ovim su autori *Wizdom*-a zaigrali na sigurnu kartu, te su na jednostavan način rešili problem rada u mreži, što je i podržano u svim funkcijama i procedurama za rad sa bazama. Jasno je da je za rad programa, pre startovanja aplikacije i *run time* modula, neophodno pokrenuti i *Btrieve* rekord-menadžer. Opšti utisak o *Wizdom*-u kao proizvodu namenjenom za brzo kreiranje jednostavnijih aplikacija potvrdile su i merenja brzine upisivanja i brisanja slogova, te pretraživanja koja su pokazala da se *Wizdom* ne snalazi najbolje već ni sa bazama od nekoliko stotine slogova. Primera radi, za kreiranje baze sa 1000 slogova, *Wizdom* je potrošio više od pet minuta! *Clipper*-u je za isti posao potrebno samo par sekundi. Ništa manje vremena nije otišlo ni na brisanje polovine slogova u bazi i pakovanje baze.

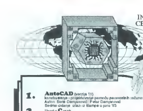
U glavnom meniju *Wizdom*-a se nalazi i opcija *Utilities* iza koje stoje alati za kreiranje programske dokumentacije, održavanje objekata, postavljanje osnovnih parametara aplikacije i drugo. Dokumentacija koju po želji korisnika Wizdom automatski generiše sadrži rečnik podataka, listu objekata i njihovu hijerarhijsku strukturu, definicije ekrana i pozicije polja za unos podataka, kao i listinge svih korišćenih programa. Kod kreiranja složenijih aplikacija ovakva programska dokumentacija je od velike pomoći. Alati za održavanje objekata se koriste za kreiranje baza i njihovo pakovanje, te za prevođenje programa, kao i za brisanje i preimenovanje objekata. Tu su i opcije za uvoz objekata iz drugih aplikacija, kao i za promenu osnovnih parametara aplikacije. U okviru parametara za svaku aplikaciju se definiše direktorijum za smeštanje datoteka, format datuma kao i polazni (*root*) i *welcome* objekat. Posebno treba istaći i postavljanje sistemskih i korisničkih lozinki. Prva se koristi za onemogućavanje neovlašćenog pristupa razvojnom sistemu aplikacije, dok je korisnička namenjena za štiti pristupa izvršnoj verziji programa. Pored globalne zaštite od neovlašćenog pristupa je moguća i za svaki pojedini objekat. Prava poslastica je mogućnost zadavanja datuma posle kojeg će program odbijati da radi, prvenstveno namenjena za demo-verzije programa.

Zbog pomenutih nedostataka, *Wizdom* je upotrebljiv samo za kreiranje aplikacija sa jednostavnim ažuriranjima manjeg broja podataka. Nadamo se da će neka sledeća verzija rešiti navedene probleme koji, na sreću po *Wizdom*, ne potiču iz odlično postavljene koncepcije već iz nevešte realizacije.

Korisna adresa:

Ei Bull HN
Stanka Popovića 41, 18000 Niš
Tel. 018/64-760, 61-781, Fax. 47-526

INSTITUT ZA NUKLEARNE NAUKE "VINČA"
CENTAR ZA PERMANENTNO OBRAZOVANJE
11000 Beograd Nemanjina 4/X
Telefoni 011 / 683-390, 682-486, 641-155 / 107, 181
Telefax 011 / 682-486



BIBLIOTEKA INFORMATIKA

1. **AutoCAD** (verzija 11)
konstruisanje i projektovanje pomoću personalnih računara
Autori Boris Damjanović Petar Damjanović
Sedmo izdanje izlazi iz štampe u junu '93

2. Uvod u **C** jezik
Autor Vladan Vujičić
Četvrto izdanje 1991 latinica 330 strana format B-5

3. Primena programa **SYMPHONY** na personalnim računarima
Autor Dragan Pantić
Treće izdanje 1990 latinica, 232 strane format B-5

4. **OS / 2** - vodič za korisnike
Autor Zorica Jelić
Drugo izdanje izlazi iz štampe u decembru '93.

5. **VENTURA** računarsko izdavaštvo
Autor Predrag Davidović
Treće izdanje izlazi iz štampe u decembru '93

6. **FORTRAN 77**
standard sa dopunama za personalne računare
Autori Vlajko Kocić i Zoran Konstantinović
Treće izdanje 1993. latinica 439 strana format B-5

7. **UNIX** vodič za korisnike
Autor Zorica Jelić
Treće izdanje 1991 latinica 436 strana format B-5

8. Primena programa **FRAMEWORK III** na personalnim računarima
Autor Dragan Pantić
Drugo izdanje 1991 latinica 330 strana format B-5

9. **Programski alati u matematici MathCAD, GRAPHER, EUREKA**
Autor Ante Ćurin
Drugo izdanje izlazi iz štampe u decembru '93

10. **QUATTRO PRO 4**
Autor Dragan Pantić
Drugo izdanje 1993 latinica 361 strana format B-5

11. **DOS UKRATKO** (verzije 3.3, 5.0 i 6.0)
Autor Dragan Pantić
Treće izdanje izlazi iz štampe u junu '93

12. Vodič za **VAX / VMS**
Autori Tamaš Kerepeš, Zvonko Orlović i Saša Matijević
Prvo izdanje 1990 latinica 532 strane format B-5

13. Primena programa **EXCEL** na personalnim računarima
Autor Dragan Pantić
Prvo izdanje 1991 latinica, 276 strana format B-5

14. **UNIX** vodič za programere
Autor Zorica Jelić
Prvo izdanje 1991 latinica 346 strana format B-5

15. **WINDOWS 3.1**
Autor Dragan Pantić
Drugo izdanje 1993 latinica 349 strana format B-5

16. **PRIMAVERA**
upravljanje projektima uz pomoć računara
Autori Jaroslav Uroševć i Jelica Drašković Ostojić
Prvo izdanje 1991 Latinica 368 strana format B-5

17. **dBASE III+** priručnik
Autor Miodrag Filipović
Prvo izdanje 1991 latinica 254 strane format B-5

18. **Osnovi informaciologije i informacione tehnologije**
Autor Ljubomir Đaković
Prvo izdanje 1991 latinica 542 strane, format B-5

19. **LOTUS 1-2-3** (verzije 3.0 i 3.1)
Autor Dragan Pantić
Prvo izdanje 1991 latinica, 416 strana format B-5

20. **dBASE IV** priručnik
Autor Ljubomir Lazić
Prvo izdanje 1991 latinica 310 strana, format B-5

21. **WORDPERFECT** (verzija 5.1)
Autori Dragan Pantić i Nada Pantić
Prvo izdanje 1991 latinica 304 strana format B-5

22. Programiranje u **CLIPPER-u 5.01**
Autor Alimpije Veljović
Drugo izdanje izlazi iz štampe u junu '93

23. **FoxPro**
Autor Dušan Čašić
Prvo izdanje 1992 latinica, 403 strane format B-5

24. **Uvod u strukture podataka**
Autor Miroslav Jocković
Prvo izdanje 1992 latinica 356 strana format B-5

25. **ORACLE** (verzija 5) arhitektura i administracija
Autor Vladimir Mićković
Prvo izdanje 1992. latinica 160 strana, format B-5

26. **WORD za WINDOWS**
Autori Dragan Pantić i Nada Pantić
Prvo izdanje 1992. - latinica 320 strana format B-5

27. **MS WORKS** upravljanje poslovnim procesima
Autori Mirjana Nikolić i Miroslav Nikolić
Prvo izdanje 1992 290 strana format B-5

28. **PASCAL**
standard sa dopunama za personalne računare
Autori Zoran Konstantinović i Slobodan Simić
Prvo izdanje 1992 latinica 401 strana format B-5

29. **OPERATIVNI SISTEMI**
MS DOS 3.30 do 6.0 NDOS 6.0 i 6.01 DR DOS 6.0
Autori Dragan Cvetković i Radica Cvetković
Prvo izdanje 1993 latinica 286 strana, format B-5

30. **LOTUS 1-2-3** (verzija 2.3) i modeli za poslovno odlučivanje
Autori Mirjana Nikolić i Miroslav Nikolić
Prvo izdanje 1993 latinica 227 strana, format B-5

31. **PC PAINTBRUSH IV PLUS (WINDOWS 3.1 APPLICATION)**
Autor Miodrag Đekićević
Prvo izdanje 1993 latinica 149 strana format B-5

32. **UVOD U SISTEMSKO PROGRAMIRANJE III**
Autor Dragan Veličković
Prvo izdanje 1993 latinica 618 strana format B-5

PORUDŽBENICA - na liniji uz redni broj knjige upisati broj poručenih primeraka

1__ 2__ 3__ 4__ 5__ 6__ 7__ 8__ 9__ 10__ 11__ 12__ 13__ 14__ 15__ 16__ 17__ 18__ 19__

IME I PREZIME (Naziv preduzeća) ____________________
ULICA I BROJ ____________________
BROJ POŠTE I MESTO ____________________
TELEFON ____________________

"PERSONALNI računari", juni 1993.

20__ 21__ 22__ 23__ 24__ 25__ 26__ 27__ 28__ 29__ 30__ 31__ 32__

Porudžbenicu pošalji sa petim primerkom uplatnice (informacije o cenama svakog radnog dana od 8 do 20 sati).
Uplata se vrši na žiro račun.
INSTITUT "VINČA", LABORATORIJA 060, broj 60803-609-698
Troškove isporuke knjiga snosi poručilac i namiruje ih unapred zajedno sa uplatom za knjige

PHOTOSHOP OTVARA PROZORE

Adobe Photoshop for Windows donosi preokret na vrhunsko tržište za obradu slika. Ako tražite unapređene alatke, specijalne umetničke efekte ili prilagodljive kistove – sve ćete naći u ovom Macintosh klasiku koji sada radi i u *Windows*-u.

Adobe Photoshop for Windows je alatka bez koje se prosto ne može zamisliti profesionalna obrada slika – reč "profesionalna" treba posebno naglasiti. Ovaj paket po ceni od 895 dolara je namenjen širokom spektru korisnika – ilustratorima, fotografima, video-umetnicima, retušerima, tehničkim urednicima – ali je suviše ozbiljan da bi služio za hobi ili stereotipne poslovne prezentacije. Uz tu napomenu, ova gotovo savršena *Windows* verzija klasičnog Mac programa zaslužuje da se zaspe pohvalama.

Temelj svakog programa za obradu slika čini njegov pristup svakodnevnim poslovima, kao što je čišćenje loše skeniranih slika ili poboljšanja kvaliteta štampe. Alatka *Photoshopa* su izuzetno moćne. Na primer, dijalog-boks *Levels* vam omogućava da pripremite fotografiju za štampu, određujući koji će delovi biti osvetljani, a koji osenčeni.

Nova opcija *Variations* nudi da podesite osvetljaj i jasnoću slike, dok točak s bojama nudi izbor jedne od šest nijansi i dva stepena osvetljanja, tako da se za čas podese ukupne vrednosti slike.

Međutim, čistunci će verovatno ostati verni tradicionalnim filterima, a to su filteri za osvetljenje i kontrast (*Brightness and Contrast*), za ujednačenja (*Equalization*) i za jačinu i zasićenost boje (*Hue and Saturation*). Možete koristiti bilo koju od ovih opcija za podešavanje tonova, bez obzira na model boja u kojem radite – siva skala, RGB, CYMK, dvoton ili LAB. Ovo je posebno važno za vrhunske profesionalne korisnike, jer pomoću *Photoshopa* mogu skenirati i modifikovati 32-bitne CYMK slike bez ikakvog gubitka informacija. Konverzija iz jednog režima u drugi se obavlja trenutno, dok je u drugim programima za editovanje slike potrebno proći kroz mukotrpan proces prevođenja.

Ukoliko vam nije dovoljno baratanje histogramima i gama-krivim, posegnite za zbirkom umetničkih kistova i alatki za retuširanje koje nudi *Photoshop*. Mogućnost da svakoj pojedinoj alatki odredite veličinu, gustinu i stepen antialiasinga znači da je kontrola u potpunosti u rukama korisnika. Mada *Photoshop* nije prevashodno program za crtanje, ima izvanredne umetničke crte, kao, na primer, sposobnost mešanja novih boja ili simuliranja poteza kistom kada boja postepeno nestaje.

Photoshop ima i funkcije za rad sa tekstom – tekst se, istina, ne može editovati na ekranu, ali dijalog-boks pruža mnoštvo opcija za formatiranje, pa i izbor između Type 1 i True Type fontova, veličina slova (od 4 do 360 tačaka), razmaka između redova i slova, kao i kontrolu uravnanja teksta.

Među novim filterima je čitav niz specijalnih efekata – *Ripple* (nabiranje), *Twirl* (obrtanje), *Lens Flare* (promjena objektiva), *Solarize* (predugo eksponiranje). Mogućnosti ovih filtera uz rotaciju i iskošenja slike rastu eksponencijalnom progresijom kada se uzmu u obzir sva sofisticirana funkcije za selekciju. Ako kažemo da alatke za selekciju mogu biti četvrtasta, elipsasta, u obliku lasa i čarobnog štapića, još uvek nismo ništa rekli o njihovoj stvarnoj moći.

Za razliku od drugih *Windows* editora slika, kod *Photoshopa* postoji jasna distinkcija između prostora za selekciju, maske i staze. Možete ih pretvarati jedno u drugo, tako da prostor za selekciju jednostavno postaje maska ili staza postaje prostor za selekciju.

Ako vam je potreban precizan ali potpuno slobodan oblikovan prostor za selekciju, možete se poslužiti alatkom u obliku pera i nacrtati oblik (ili stazu) pomoću Bezier krivih, a zatim ga pretvoriti u oblast za selekciju. Isto tako, ako želite da primenite neki efekat, ali sa promenljivim intenzitetom, opcija **Quick Mask** će pretvoriti trenutnu selekciju u 8-bitnu masku koja se može editovati.

Upravljenje mnogobrojnim maskama i objektima sastavljenim od Bezier krivih je lakše nego ikad, zahvaljujući dvema novim paletama – *Channels* i *Paths*. *Channels* obezbeđuje brz pristup do 16 različitih kanala, a svaki služi za skladištenje jedne komponente boje (shodno tome, za RGB podatke su potrebna tri kanela, za CYMK četiri) ili jedna 8-bitne maska.

Photoshop za *Windows* u mnogo čemu emulira sopstvenu Mac verziju. Datoteke su bi namo kompatibilne na različitim platformama. Osim toga, *Photoshop* ostvaruje fantastičnu spregu sa programom *Adobe Illustrator* rasterizujući "u letu" (*on the fly*) vektorske EPS slika.

Podražavanja Maca ima i svoju lošu stranu – neki filteri zahtevaju toliko intenzivan rad kompjutera da se slika u potpunosti čuva u RAM-u za vreme primene određenog efekta, tako da čak i sa 16MB RAM dobijate poruku o nedostatku memorije.

U svakom drugom pogledu, performanse *Photoshopa* su zadivljujuće. Pod uslovom da imate brzu mašinu, *Photoshop* će biti u stanju da drži korak sa svojim najbržim konkurentom – *Aldus PhotoStyler*. Ali to nije sve – brzina *Photoshopa* može dostići još neslućene razmere: optimizovan je za rad sa *Pentium* čipom, a otvorena arhitektura znači da će prihvatati novu vrstu DSP ploča (*Digital Signal Processor*) koje će ubrzati operacije.

Photoshop odgovara gotovo svakom radnom okruženju. Izvanredne opcije za uvoz i izvoz podataka podržavaju mnoge tipove rastera – RGB, PCX, Pixar, Scitex, Targa, TIFF i EPS, JPEG. Direktno prima skanirane slika (preko *Adobe Plug-in* podrške za *Twain* interfejs za skeniranje) i generiše četvorobojne separacije.

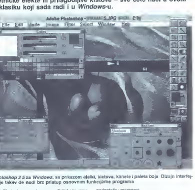
Sve ove pohvale ne znače da se vaš izbor svodi na *Adobe Photoshop* ili ništa. Alternativni programi *Picture Publisher* (Micrografx), *Foto-Finish* (ZSoft) ili *Image Assistant* (Caere) se daleko lakše savladuju i drastično su jeftiniji. Ali, ako želite da vaš elektronski studio zasnovan na PC računaru dobije sposobnosti jednog Mac-a, *Photoshop* je program bez kojeg apsolutno nećete moći.

Cena: 695 dolara

Zahtevi: 4MB RAM, DOS 5.0, *Windows 3.1*, PC 486; preporučuje se 8 MB RAM i 24-bitni grafički adapter

Izvor: *PC Magazine*

Prevela: Ranka Jovanović



***Photoshop 2.5* za *Windows*, sa prikazom alatki, kistova, kanela i paleta boja. Dizajn interfejsa je takav da nudi brz pristup osnovnim funkcijama programa**

JEZIK ZA PO KUĆI

Firma koja je napisala jedan od prvih bejzika na svetu, a kasnije napravila jedini ozbiljan grafički operativni sistem za PC računare, izbacila je na tržište čudesnu kombinaciju ta dva paketa – firma je „Microsoft", operativni sistem je (naravno) *Microsoft Windows*, a bejzik koji je tema ovog prikaza je *Visual Basic 2.0*

Dušan Popović

Microsoft Visual Basic je jezik koji je kada se pojavio (u verziji 1.0) obećavao nešto krajnje kontraktorno – lako pisanje *windows* programa. Kako smo već navikli da „lako" ne ide uz „windows", nismo odoleli da ne isprobamo tu čudnu kombinaciju bejzika i objektnog programiranja. Pokazalo se da i ta rana verzija nudi rešenja koja nisu viđena od tada, omogućujući da se programi pišu „za pet minuta".

Ono što je omogućilo takvu lakoću pravljenja aplikacija pod *Windows*-ima je potpuno nov pristup samom pisanju. Dok je u raznim SDK paketima proces pisanja programa strogo razdvojen na crtanje i definisanje resursa (menija, ikone i dijaloga), u *Visual Basic*-u je sve na jednom mestu – kako aplikacija raste, postepeno se dodaju novi resursi, bez brige da li je sve na početku dobro isplanirano.

Sve što smo od sada rekli moglo se odnositi na obe verzije *Visual Basica*. Ono što smo očekivali od verzije 2.0 je da će potencijalni bagovi biti očišćeni i dodate nove mogućnosti.

NIŠTA GLOMAZNO

Visual Basic nije paket od koga će vam se saviti kičme dok ga budete prenosili sa jednog mesta na drugo. Sam paket izgleda otprilike kao četvrtina ili petina pakovanja normalnih C kompajlera. O težini da i ne govorimo – dve knjige koje idu uz *Visual Basic* se ne mogu meriti sa metrima literature za *Borland* ili *Microsoft C*. Malo je čudno što poredimo jedan bejzik kompajler sa dva C kompajlera, ali svi služe istoj svrsi, pisanju programa pod *Windows*-ima. Još je čudnije što se u već pomenute dve knjige nalaze sve potrebne informacije za pisanje aplikacija koje veoma profesionalno izgledaju i koje imaju sve *Windows* odlike – menije, prozore, dijaloge. Sve to govori o jednostavnosti paketa sa kojim imamo posla.

Instalacija traje relativno kratko, ali zato ume da zaboli, posebno ako imate manji hard disk. Radi se o tome da je za instalaciju potrebno oko 15 megabajta na disku. 15 megabajta za običan bejzik?! Kuda ide ovaj svet? Ipak – ukoliko ovaj paket može da zadovolji većinu vaših potreba, i ako ga uporedimo sa zauzećem diska nekog novijeg C kompajlera, možemo reći da nije sve tako crno.

I na kraju, da još jednom spomenemo literaturu. Uz paket idu dve knjige: *Language reference* (spisak funkcija, procedura, objekata...) i *Programmer's guide* (klasično uputstvo za upotrebu). Uputstva su sasvim standardna, ali zadovoljavajuća (pomalo nam nedostaje humor Borlandovih autora). Knjige nećete tako često otvarati, zahvaljujući jednostavnosti paketa, ali i izuzetno dobro napisanom help sistemu koji pod *Windows*-ima lepo izgleda, pregledan je i lako se koristi.

UKUS BEJZIKA

Visual Basic je jezik koji dozvoljava postupno učenje i lagan prelazak sa nekog od standardnih bejzika (*qbasic*-a ili *gwbasic*-a, na primer) preko komplikovanijih tehnika do pisanja prave *Windows* aplikacije. To znači da možete gotovo svaki bejzik program koji ste ranije napisali (a kod svakoga se nakupio poveći broj takvih programa tokom godina) da jednostavno kompajlirate ili startujete iz *Visual Basic* okruženja. Kažemo gotovo svaki, jer se sasvim logično ograničenje odnosi na DOS programe koji imaju posla sa grafikom – ne samo da je rad sa grafikom pod *Windows*-ima drugačije rešen, nego i DOS narečja bejzika imaju međusobno nekompatibilne komande. Kada startujemo neki „normalni" bejzik program, on se ponaša kao u tekst-modu – ceo prozor služi za ispis i unos teksta (naravno, pomoću običnog printa i inputa), a ako zanemarimo okvir prozora, još uvek nema ni traga od *windows* elemenata.

Ovaj bejzik, kao i svaki moderniji organizovan jezik, ceo program čuva podeljen po procedurama. Taj pristup nije nov – na primer, može se videti kod *qbasic*-a koji je sastavni deo DOS-a. U *Visual Basic*-u je podela po procedurama još jasnija, jer su one i fizički odeljene – svaka procedura je vezana za neki objekat, odnosno za njegovu akciju. Iako ovi termini mogu zvučati zbunjujuće, radi se o vrlo jednostavnoj i prirodnoj upotrebi procedura, a samo snalaženje sa takvom organizacijom koda se savladuje za desetak minuta.

Po startu *Visual Basic*-a, korisnik se suočava sa nekoliko otvorenih prozora. Ako ste već radili sa prethodnom verzijom ovog paketa, sve će vam biti jasno, jer su prozori pretrpeli minimalne izmene. Za korisnike koji se prvi put sreću sa *Visual Basic*-om, kratko ćemo pomenuti „šta se nudi" i šta čemu služi.

Na samom vrhu ekrana se nalazi glavni meni *Visual Basic*-a, kao i *toolbar* sa prikladno odabranim ikonama koje kontrolišu sve operacije vezane za stvaranje *basic* aplikacije pod *Windows*-ima. Tu se mogu naći opcije za snimanje i učitavanje programa, njegovo startovanje ili kompajliranje, kao i za uključivanje i isključivanje nekog od ostalih prozora.

Sa leve strane ekrana se nalazi vertikalni *toolbar* - izbor raspoloživih elemenata. Tu se nalazi spisak svega što nam može zatrebati kao detalj u nekom od dijaloga aplikacije – od natpisa i unosa, do objekata za izbor diska ili direktorijuma. Samim korišćenjem elemenata ćemo se više pozabaviti malo kasnije.

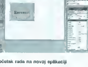
Desno se nalazi *Project* prozor, odnosno *Visual Basic*-ov *make-file*. Za početak je veoma važno shvatiti organizaciju koju nameće pisanje komplikovanijih aplikacija pod ovim jezikom: svaka aplikacija može imati veći broj takozvanih „formulara". Svi oni sadrže dve vrste elemenata – same objekte (unose, dugmad i slično) i procedure vezane za njih. Kada pišemo neki komplikovaniji program, sve nove formulare uključujemo u projekat, da bi *Visual Basic* znao šta sve treba da uključi u konačnu aplikaciju.

Dole se nalazi prozor *Properties*, spisak raspoloživih „osobina" nekog elementa. Ovaj prozor nam služi da u toku kreiranja neke aplikacije precizno odredimo izgled i ponašanje svakog od elemenata – boju, font i veličinu slova, izgled.

Prozor *Menu design* se inicijalno ne vidi. Njega ćemo pozivati kada želimo da naše aplikacije dobiju menije.

Na sredini se može videti prozor *Form*, odnosno prvi prozor naše buduće aplikacije. U *Form* prozoru ćemo, postepeno dodajući elemente i vezujući kod za njih, dobiti prvu *Windows* aplikaciju.

Možda ste primetili da do sada nismo pomenuli prozor u kojem se piše program. Ipak, svako skraćenje priče ne bi bilo u skladu sa idejom ovog jezika – zato idemo redom, upoznajući stvari onako kako ih je „Microsoft" organizovao.


Početak rada na novoj aplikaciji

ORGANIZACIJA

Već smo spomenuli formulare. Formulari su osnova svega. Da bismo uopšte napisali neki *Visual Basic* program, moramo imati bar jedan (osnovni formular). Šta je u stvari formular? To je običan *windows* prozor, potpuno prazan na samom početku pisanja novog programa. U prozoru *Properties* se vide sve njegove „osobine", jer je ceo prozor za sada jedini selektovani element. Sada polako možemo dodavati elemente u formular. Recimo da želimo da prozor naše aplikacije ima samo jedan taster. To ćemo jednostavno postići – prvo je potrebno iz spiska raspoloživih elemenata izabrati ikonu na kojoj je nacrtan taster, a zatim mišem „razvući" taster bilo gde u okviru formulara, na željenim koordinatama i u željenoj veličini. Svakom elementu je moguće dodati procedure i tu već dolazimo od **akcija**.

Akcije su jednostavno procedure koje se izvršavaju kada se objektu „dogodi" poziv te akcije. O čemu se radi? Recimo da imamo objekt „dugme". Tom objektu možemo nešto „uraditi". Na primer, na dugme možemo kliknuti, možemo mu predati fokus (kada pomoću tab tastera prenesemo okvir na taj taster) ili mu ga oduzeti. Za svaku od mogućih akcija postoji po jedna procedura. Procedure su na početku prazne, a na nama je da dodamo željeni kod sa-



Kako raste program

mo u neke od njih. Ako uzmemo prethodni primer (samo jedan taster na formularu) i želimo da program završi sa radom posle klika na taster, postupak je sledeći (pretpostavljamo da su formular i taster već nacrtani): prvo duplim klikom na taster dobijamo prozor za unos koda (konačno?), zatim iz spiska raspoloživih objekata izaberemo onaj koji treba da „reaguje". U našem slučaju na raspolaganju su nam samo **form1** i **commend1**. Izaberemo **command1** (što je *Visual Basic*-ov interni način za numerisanje tastera). Zatim iz spiska mogućih akcija izaberemo onu na koju objekat treba da reaguje. Iz spiska biramo „click" akciju koja se poziva kada se klikne na objekat. Na delu prozora predviđenom za unos koda formirao se naziv procedure – COMMAND1_CLICK (). Imena procedura jasno govore o tome za koji objekat i za koju akciju su vezane. Tada jednostavno unesamo komandu END u proceduru. To je sve.

Sada možemo da isprobamo program koji smo napisali. Taster F5 služi za start programa iz *Visual Basic*-a. Po startu se na ekranu jasno vide prozor i naš taster, baš onakvi kakve smo ih kreirali. Sada jedan klik na taster vodi iz programa – time se poziva naša procedura, a END završava program.

Ovim smo odradili najjednostavniji slučaj pisanja programa, jer broj elemenata po formularu nije ograničen, a formular ne mora da bude samo jedan – programi mogu dosta da narastu.

OBJEKTNE KOMPLIKACIJE

Da bi sve bilo mističnije, ili možda zato što termini za opisivanje objekata još uvek nisu usaglašeni od proizvođača do proizvođača i od jezika do jezika, „Microsoft" za rad sa elementima *Visual Basic*-a koristi reči kao što su **properties** ili **method**, što smo mi preveli kao osobine i funkcije. U stvari, ceo *Visual Basic* je na pola puta do pravih objekata i možemo da kažemo da nam je drago što je tako – svako nepotrebno po...svaku...cenu komplikovanje i upetljavanje pravog objektnog programiranja u jedan tako jednostavan jezik bi imalo kontraefekte – odbijalo bi zdrav deo programerske populacije koja se ne smeška blaženo pri samom pomenu objekata. Svako ima pravo da ne bude oduševljen objektnim programiranjem i da mu prilazi sa oprezom. Objekti nude lepa rešenja, ali ne na svakom mestu, ne za svaki problem i ne po svaku cenu.

Na naše veliko zadovoljstvo, ovde se ne radi o pravim objektima, iako ima objektnih elemenata. Na primer, svaki objekat ima određene elemente koji su vidljivi ostalom delu aplikacije. Tako „u trku", u toku rada programa, možemo menjati natpise na tasterima ili osvežavati spisak fajlova u nekom prozoru, jer su to



Vezivanje koda za objekt

„javni" delovi objekta. Takođe, svaki objekat ima određeni broj metoda, odnosno funkcija preko kojih mu se možemo obratiti. Time je stvorena fina kontrola toka programa, uz potpuno poštovanje *message driven* načina rada sa *Windows*-ima. Tako, na primer, pomoću komande **form2.show()** i **form2.hide()** predajemo ili ukidamo fokus narednom prozoru aplikacije. Sve to liči na objektno programiranje, ali ipak nije.

Pri pisanju programa (a posebno ukoliko vam objekti nisu u krvi) možete na objekte slobodno zaboraviti. Jednostavno postoje funkcije koje imaju svoje vlasnike (elemente i akcije) i koje će se izvršiti kada se desi određeni događaj. Takođe, svi elementi imaju određena javna osobine koje se zadaju u toku kreiranja programa, ali se kasnije mogu menjati iz samog programa, prilagođavajući time izgled aplikacije. Sve je stvar terminologije – možete koristiti termine koji zvuče objektno, ili nekako drugačije – kako vam je lakše.

PISANJE ILI CRTANJE?

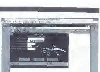
Pisanje programa pod *Visual Basic*-om je neobično, to je sasvim jasno. Kada se malo naviknete na takav način razvoja aplikacija, čudno će vam biti kako se niko nije setio nečeg sličnog. Svaki *Visual Basic* program se razvije postupno – element po element, redom se dodaju potrebne procedure i program polako raste. Prednosti ovakvog rada su ogromne – dok u C-u morate na početku razvoja nove aplikacije da znate, ako ne baš sve, a ono bar najvažnije elemente koje ona mora da ima, i da ih definišete pre samog početka pisanja programa (obično u nekom od editora resursa koji idu uz *Borland* ili *Microsoft C*). Za razliku od toga, program u *Visual Basic*-u ne mora odmah da ima sve, pa čak ne morate tačno znati kako će i šta će program na kraju raditi. Svi elementi se biraju iz *toolbar* linije sa alatima, dinamički premeštaju po formularu i menjaju veličinu, a zatim se njima dodaju potrebne akcije.

Sve u svemu, imamo krajnje interaktivan način pisanja programa, nasuprot krutom tradicionalnom definisanju resursa.

Ovakav način pisanja programa obezbeđuje još jednu važnu stvar – krajnju opuštenost u radu, bez brige o pogrešnom planiranju. Sve elemente je moguće lako promeniti ili izbrisati u bilo kojoj fazi razvoja, što prosto tera programera da eksperimentišu dok ne budu potpuno zadovoljni rasporedom elemenata i funkcionisanjem cele aplikacije.

DUBOKI KORENI

Visual Basic je paket koji će sigurno uspeti da zadrži svoj deo tržišta. Postoji nekoliko pra-



Ozbiljni programi u bejziku

vih odluka koje garantuju da ovaj paket može da se bori sa konkurentskim rešenjima za pisanje aplikacija pod *Windows*-ima. Pre svega, bejzik je jezik sa kojim se svako morao sresti, od Spectruma od velikih sistema. Taj jezik se takođe malo menjao tokom godina – vremenom su dodate procedure i opciono deklarisanje varijabli, ali je to, u osnovi, isti onaj jezik koji je legendarni Bil „Microsoft"-a napisao pre mnogo godina. Ukratko, ko je ikada pisao neki program u bejziku, znače da se snađe i u ovoj verziji.

Dodatne funkcije koje su neophodne za pisanje *Windows* aplikacija su prirodno uklopljene u standardni bejzik, a od programera se čak i ne zahteva da odmah piše prave *Visual Basic* programe – za početak je dovoljan i običan **print**.

Visual Basic vuče korene od pre mnogo godina, pa zato mora da ima i poneku manu. Na primer, brzina generisanog programa se ipak ne može porediti sa nekim C kodom, program je potencijalno bagovitiji, jer se greška može pojaviti tek pri radu programa, a mašina se ne može kontrolisati do poslednjeg bita kao u C-u.

Sa druge strane, postoji veliki broj lepih rešenja koja su u bejziku normalna stvar, a u paskalu ili c-u slabije podržana. Na primer, rad sa stringovima je tako rešen da lako razmazi čoveka. Malo vredi bilo kakva podrška stringovima iz c-a ili paskala, kad u bejziku u sred programa možemo napisati „jedan"+"dva". O prednostima bejzika nad ostalim jezicima možemo pisati još puno, ali je to nepotrebno – ionako sigurno govorite bejzik i znate o čemu je reč.

Teško je bilo odlučiti kakvu ocenu da damo ovom paketu i kojem delu tržišta on pripada – ozbiljnom, u kojem se pišu ozbiljne aplikacije, ili onom drugom, gde je potrebno nešto uraditi na brzinu, za po kući. Mislimo da je ova verzija *Visual Basic*-a nešto stvarno vredno i da joj treba pokloniti pažnju. Rešenja koja nudi ovaj jezik su njegova privilegija – toliko su laka za primenu da mislimo da je ovo prvi i jedini jezik kod kojeg pisanje *Windows* aplikacija ne znači istovremeno i glavobolju za programera. Poboljšanja koja su došle u ovoj verziji *Visual Basic*-a su pomak ka profesionalnijem delu tržišta.

Naša ocena ovog paketa je vrlo visoka, uz nadu da će još neki proizvođač (Borland?) ponuditi slično organizovane jezike, stvarajući time zdravu konkurenciju. *Visual Basic* je sve manje igračka, a sve više ozbiljan alat.

Korisna adresa:

CET, 11000 Beograd, Skadarska 45
Tel. 343-043

GOSPODAR DESKTOPA

Sa paketom *PC Tools for Windows* „Central Point Software" je od prve postigao pun pogodak. Po ceni od 179.95 dolara, nova verzija uključuje i *Microsoft Windows* desktop, mnogo fleksibilniji i prilagodljiviji potrebama korisnika od samog *Program Manager*-a ugrađenog u *Windows*. Ipak je po dizajnu dovoljno sličan *Windows*-u, tako da nije potrebno mnogo vremena za navikavanje.

Softver sadrži ČITAV niz izvanredno moćnih alatki za održavanje diska i upravljanje datotekama: *batch* jezik koji jednostavno makro beleženje kombinuje sa efikasnom funkcijom za programiranje, poznate i proverene programe „Central Point"-a za *back-up* i borbu protiv virusa, i još mnoštvo praktičnih opcija kao što su pružanje informacija o sistemu, štednja prostora na ekranu i editor ikona.

DESKTOPI I OFISI

PC Tools ima *drag and drop* desktop koji može biti krajnje oskudan, ali i veoma bogat i raznolik, zavisno od želja korisnika. Možete ukloniti liniju sa alatkama (*toolbar*), naslovnu liniju, pa čak i meni u gornjoj liniji i tako dobiti potpuno čist ekran. Pritiskom na drugi taster miša u svakom trenutku se dobija „brzi meni" (*quick menu*) opcija koje se odnose na ikonu, meni, prozor ili desktop koji se nalazi ispod kursora.

Pri instalaciji ovaj paket kreira niz desktopa, svaki sa različitim skupom aplikacija i datoteka, i u različitoj boji. Iz jednog u drugi se prelazi pomoću *hot key* kombinacija ili klikom na minijaturnu mapu koja prikazuje sve desktop-e, a pri prelasku možete premestiti i bilo koji prozor. Aplikacije koje ostavite otvorane pri napuštanju *Windows*-a automatski se ponovo otvaraju na istom mestu kada ponovo startujete program.

Jedan ili više desktopa se mogu organizovati u grupe koje se nazivaju ofisi (*office*) a istovremeno se može otvarati više ofisa. Za razliku od virtuelnog desktopa iz paketa *hDC Power Launcher*, ovde može biti aktivan samo jedan desktop. Ne možete pomoću *Alt-tab* prelaziti iz programa u program u različitim desktopima, niti možete istovremeno na ekranu imati pola jednog desktopa, a pola nekog drugog.

PC Tools vam pruža mogućnost da smestite ikone za programe i datoteke bilo gde, u bilo koji desktop – pa i sve u jedan desktop, kao i *Norton Desktop for Windows* – ili da ikone svrstate u grupe, kao u *Program Manager*-u. Grupe mogu biti prikazane na uobičajen način, kao ikone, ili pak kao kutije za alatke bez potpisa, kao lista ikona uz koju ide opis, ili kao sažeta lista u kojim se ne pominje ništa osim imena.

Svaka grupa može imati različit razmak između ikona, boju pozadine, lozinku i još desetine drugih opcija. Desktop i sva druga alatke u paketu imaju menije koji se mogu prilagoditi potrebama. Ni jedan paket vam neće ponuditi podešavanja do tako sitnih detalja, niti će vam dozvoliti da provedete toliko vremena prilagođavajući sva što se vidi na ekranu svojim potrebama, ili potrošite toliko memorije i resursa da biste pravili mnoštvo desktopa i otvarali mnogobrojne aplikacije. Srećom, desktop se može privremeno zatvoriti, tako da se po potrebi oslobađa memorija.

PC Tools desktop se učitava i radi brže nego *Norton Utilities*, a jedina ozbiljna mana mu je nedoslednost. Ako imate dve *Excel* tabele u dva desktopa, program je dovoljno pametan da zna da treba da otvori samo jednu instancu *Excel*-a, ali ne ume da natera *Excel* da prelazi iz tabele u tabelu kada vi prelazite sa desktopa na desktop. Osim toga, automatski će se vratiti u ranije pokrenutu aplikaciju pri ponovnom startovanju *Windows*-a samo ako ste prvobitno aplikaciju pokrenuli preko desktop ikone ili grupe, a ne iz fajl-menadžera ili klikom na komandu **Run.**

MENI ZA SVAČIJI UKUS

U liniju s alatkama (*toolbar*) i menije se može uneti svaki program, datoteka ili komanda, uključujući i *batch* komande napisane ili zabeležene u skriptu. Ali, svaki uslužni program sadržan u paketu ima svoju zbirku ugrađenih alatki koja ne može razmenjivati sa drugima. Na primer, fajl menadžer ima alatku koja klikom miša otvara DOS prozor, ali ako sličnu alatku želite u liniji alatki desktopa, moraćete je sami kreirati.

Za razliku od *Norton Desktop*-a (nedavno se pojavila dograđena verzija 2.2 sa unapređenim mrežnim opcijama i podrškom za DOS 6.0), *PC Tools* upravljanje datotekama i programima obavlja pomoću posebnih aplikacija. *PC Tools* fajl menadžer po izgledu i po načinu rada podseća na dorađenu verziju *Windows File Manager*-a. Ima prozore za brz pregled datoteka koji prikazuju fontove korišćene u dokumentu i čak automatski proračunavaju numeraciju paragrafa.

Prozori za pregled u *PC Tools*-u znaju da se zaguše u prilikama u kojim se *Norton* snalazi bez problema, ali zato ima stvari koje *Norton* ne ume, a *PC Tools* obavlja bez greške. *PC Tools* ima 20 alatki (*applets*) koji obavljaju standardne funkcije upravljanja datotekama, a da pri tom nije potrebno otvaranje samog fajl menadžera.

Fajl menadžer vrši kompresiju datoteka u PKZIP 1.x format i prikazuje datoteke u većini ostalih komprimovanih formata. Kao kod *Norton*-a i fajl-menadžera iz paketa *Wordperfect for Windows*, *PC Tools* ima sposobnost da „skoči" na određeno ime datoteke kada ukucate dovoljno slova da ga prepozna.

ALATI ZA RAD S DISKOM

PC Tools između ostalog sadrži i program po imenu *DiskFix* koji ima jedinstvenu sposobnost obavljanja popravki na mašinskom nivou bez izlaska iz DOS-a. Tu je i defragmentor koji radi u pozadini dok se izvršavaju drugi programi. Čovek treba da ima dobre živce da bi hladno posmatrao ovaj proces, s obzirom da pozadinska defragmentacija može biti smrtonosna za vaš disk, ukoliko *Windows* ne uspe da prati promene strukture diska. Defragmentor se iz bezbednosnih razloga zaustavlja svaki put kada *Windows* pristupa bilo kojem disku u sistemu. Zatim se morate vratiti u meni da biste ponovo startovali defrager, mada se može podesiti i da se automatski ponovo startuje nakon određenog vremena.

Ako spadate u one koje su nervirali makro-rekorder i *batch* jezici iz *Windows*-a, bićete oduševljeni skript-alatkama *PC Tools*-a (*Script Tools*) koje prevazilaze praktično sva ograničenja i nedostatke konkurentskih paketa. Skript-alatke, doduše, nisu brze kao programski jezik, ali nude lakši put do unapređenih *batch* datoteka nego bilo šta dosad viđeno. „Central Point" ne otkriva ko je napisao *Script Tools*, ali program neodoljivo podseća na *Automator for Windows* koji potpisuje firma „Direct Technology".

Tu nije kraj mogućnostima *PC Tools*-a – u paketu se nalazi i sistemski konsultant koji izveštava o sistemu i predlaže kako da ga unapredite, alatka koja kreira „prinudni" disk (*emergency*), *undelete* alatka, pa zatim antivirusni softver, editor ikona i planer (*scheduler*) pomoću kojeg možete otkazati događaj ili ga samo odložiti pritiskom na *snooze* dugme. Tu je i popularni program za *backup* sa snažnom podrškom za strimere i mreže.

PC Tools nema neke desktop alatke koje ima *Norton Utilities*, npr. kalkulator ili ikona za više štampača, ali od svih *Windows* paketa nudi najkorisnije i najopsežnije alate za rukovanje desktopom, diskom i datotekama. Kao i sve zamene za desktop paket *PC Tools* je možda obimniji i složeniji nego što vam treba, ali je neosporno superioran u odnosu na originalni *Windows* i dovoljno moćan i fleksibilan da omogući da kreirate baš onakav desktop kakav ste zamislili.

Zahtevi: 4MB RAM, 3-14MB prostora na tvrdom disku, *Windows 3.1* (enhanced mode)

Izvor: *PC Magazine*

Prevela: Ranka Jovanović

Ukratko

PC Tools nudi bezbroj alatki za disk i datoteke, kao i vernu zamenu za desktop koji omogućava korisniku da prilagodi sebi rad u *Windows*-ima.

Demo-verzija *PC Tools for Windows* se može skinuti (*download*) sa PC MagNet-a – treba kucati GO ZNT EXEC sa bilo kog CompuServe prompta.

Nastavak sa strane 33

onoma što je najvažnije – u stepenu i brzini kompresije. Pri testiranju ova dva osnovna kriterijuma, *SuperStor Pro* pokazuje ispodprosečne rezultate. Sposobnost rada u *Windows*-u i dobre uslužne alatke čine *SuperStor Pro* ozbiljnim konkurentom, ali mu performanse ne dozvoljavaju da nadmaši *Stacker*-a.

XTRADRIVE 1.0

XtraDrive (Integrated Information Technology) na neobičan način postiže povećanje kapaciteta. On udvostručuje kapacitet tvrdog diska promenom *boot* staze, zamenom DOS *boot* zapisa na BIOS nivou i usmeravanjem poziva disk funkcija kroz svoj drajver i hendler interapta. Ovaj pristup ima svoje dobre i loše strane: jednostavan je za korišćenje, ali zato oporavak podataka može biti vrlo težak.

XtraDrive se izvršava pod DOS-om. U paketu nema *Windows* programa, mada softver sasvim dobro radi sa *Windows*-om. Opcije za instalaciju vam omogućuju da odaberete željeni stepen kompresije, od 2:1 do 8:1. Možete odabrati veličinu EMS keša – od jednog do četiri bloka od 16 KB, kao i opcioni keš sistemskog RAM-a iste veličine. Za testiranje je odabran jedan blok od 16 KB za keš sistemskog RAM-a, a EMS keš je isključen. Zavisno od opcija, drajver i keš koriste od 40 KB do 90 KB sistemske memorije, mada se mogu učitati u visoku memoriju.

Program kreira dva komprimovana diska čija je veličina jednaka originalnom nekomprimovanom disku. Od drajva D čiji je kapacitet 10MB, nastaju drajv D od 10 MB i drajv E od 10 MB. *XtraDrive* čuva sve originalne podatke na novom drajvu D, tako da su sve staze sačuvane.

Originalni disk sadrži nevidljivu datoteku IITV3.VOL od oko 160 KB. IITV3.VOL u stvari predstavlja datoteku sa FAT (*file allocation table*) i drajv informacijama o *XtraDrive* virtuelnom drajvu. To je samo jedna od dve kopije, druga se čuva na zaštićenom mestu izvan komprimovane particije. Ako se IITV3.VOL datoteka ošteti ili izbriše, uslužni program *XtraDrive*-a može da je oporavi.

XtraDrive nije obiman paket. Ima samo tri softverske alatke: **xdir**, koji je sličan kao DIR i daje informacije o kompresiji i raspoloživom prostoru, alatku koja je ekvivalentna komandi **chkdsk**, kao i alatku za defragmentaciju.

Zbog sasvim jedinstvenog načina modifikovanja tvrdog diska, gubitak ili oštećenje *XtraDrive* drajvera sprečava pristup celom tvrdom disku. Kod drugih paketa (npr. *Stacker* i *SuperStor Pro*), takvo oštećenje onemogućava pristup samo arhiviranom delu diska. Zato je izuzetno važno da striktno sledite korektnu proceduru pri deinstalaciji *XtraDrive*-a.

XtraDrive pruža solidnu kompresiju i brzinu. No, njegova tehnika kojom se direktno menja *boot* staza je neuobičajena. Za razliku od *Stacker*-a i *SuperStor Pro*, ovaj paket, koji se prodaje po ceni od 99 dolara, ima malo softverskih alatki i nema *Windows* interfejs. (Doduše, uskoro se očekuje verzija 2.0 sa većim brojem alatki.) Princip kompresije koji koristi *XtraDrive* je jednostavan i lak za korišćenje, ali modifikacija *boot* staze unosi nedoumice zbog kojih se ovaj paket ne može mirne duše preporučiti.

Pokazalo se da je u svemu najbolji *Stacker*: postiže najbolje performanse, visok stepen kompresije i brzinu, ima dobar korisnički interfejs i za *Windows* i za DOS, poseduje veoma informativan skup uslužnih programa, kao i dobro napisano uputstvo.

Izvor: *Byte*

Prevela *Ranka Jovanović*

TIPOGRAFIJA KROZ IGRU

U „Računarima 86" predstavili smo izuzetan program domaćeg autora Milana Velimirovića za rad sa konturnim fontovima – *Tipotek*. U tom trenutku, program je na našim prostorima imao slabu konkurenciju inostranih programa, a ono malo što nam je bilo dostupno gotovo da i nije zavređivalo poređenja sa daleko boljim *Tipotek*-om. Za poslednjih 6 meseci dosta se toga promenilo – dobili smo i bolje inostrane programe i novu verziju *Tipotek*-a – 2.01.

Zoran Životić

Za one koji se prvi put sreću sa ovim programom treba reći da se radi o programu za kreiranje i editovanje vektorskih fontova. Podržani su gotovo svi popularni formati: *Adobe Type 1* (PBF), zatim *LaserMaster* (LXO), *CorelDraw* (WFN), *Zsoft Publisher's Type Foundary* (OTL), URW i konačno *TrueType* format koji zahvaljujući *Windows*-ima sliče dominantnu ulogu na tržištu. Program radi u grafičkom okruženju koje je potpuno realizovano u samom programu. Verzija 1.0 se isporučivala na 1 disketi uz štampano uputstvo od oko 80 strana.

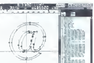
Osnovne karakteristike verzije 1.0 se mogle opisati sa dve reči – izuzetna operativnost. Već posle prvih iskustava sa programom je postalo jasno da je on delo autora koji ne samo da dobro poznaje problem već ima i iskustva u radu sa fontovima. Izbor opcija programa, način njihove realizacije, pa i niz detalja o kojima je vođeno računa, učinili su da program postane jak alat kojim se komplikovani posao izrade fonta može obaviti.

Verzija 2.01 je otišla još dalje u tom pravcu. Korisnički interfejs je i dalje zasnovan na sopstvenom grafičkom okruženju, ali je učinjen napor da se rad sa programom učini još preglednijim. Sve opcije koje su se ranije aktivirale preko ikona u nekoj vrsti horizontalnog kontrolnog panela na vrhu ekrana, sada su nešto logičnije raspoređene. Uveden je meni, u koji su prebačene opcije koje se tiču globalnog rada sa programom – snimanja i učitavanja fonta sa diska, izmena font informacije, podaci o fontu ili prelazak u neki drugi režim rada, kao što je rad sa kodnom tabelom ili parovima podsecanja. Ovaj meni se otvara posebnim dugmetom, tako da je kontrolni panel sada znatno rasterećen i skoncentrisan na opcije i režime crtanja slova. Desna površina ekrana, na kojoj se može pratiti izgled slova u popunjenom obliku, u obliku listinga elemenata koji ga čine ili u odnosu na druga slova, je takođe promenila dizajn. I sam crtež na radnom ekranu je dobio zgodan detalj – tačke kojima se opisuju kontura slova su sada numerisane na vrlo pregledan način, tako da se lako detektuju pojedini segmenti ili određuje smer crtanja.

Najkraće rečeno, sve dobre osobine verzije 1.0 su u potpunosti zadržane, opcije su preglednije raspoređene i dodat je niz novih detalja koji dalje povećavaju operativnost programa.

U poređenju sa konkurentskim proizvodom *Fontographer*-om, koji je trenutno neka vrsta referentnog programa u ovoj klasi, *Tipotek* prolazi više nego dobro. Iako sam pristalica standardizacije okruženja, te bi trebalo automatski da dam prednost *Fontographer*-u koji radi pod *Windows*-om, ipak to neću učiniti. *Fontographer* zaista jeste dobar proizvod i u nekim svojim mogućnostima nadmašuje *Tipotek*, ali bih za kreiranje kompletnog fonta ipak izabrao ovaj drugi, jer je u ovom domenu ipak operativniji, sa više mogućnosti, pa čak i brži u radu. Za poređenje filozofije koja stoji iza ova dva programa, možda je ilustrativan primer rada sa TTF fontovima.

Fontographer ne postavlja nikakva ograničenja pri konverziji različitih formata, pa tako možete konvertovati bilo koji od ulaznih formata u izlazni *TrueType* format koji je sada posebno zanimljiv. Sa druge strane, *Tipotek* razdvaja drajvere za ulaz i izlaz, pa tako može da učita TTF font ali ne može i da ga snimi u tom formatu. S obzirom da TTF format koristi drugačije elemente od uobičajenih za formiranje kontura slova (koristi se kvadratni B splajnovi umesto manje više



standardnih kubnih Bezijerovih krivih), postavlja se problem kako ovu konverziju izvršiti a da ne dođe do gubitka originalnog oblika slova. *Fontographer* čini što može, pa je rezultat promenljivog kvaliteta – na većim veličinama gomilanja dodatnih tačaka se ne primećuje suviše, dok je kod manjih veličina, posebno u ekranskom prikazu, gomilenje toliko da je font gotovo neupotrebljiv. Autor *Tipotek*-a je takođe pokušao da reši ovaj problem – rezultat nisam imao prilike da vidim, ali je odluka autora da ova konverzija ne zadovoljava ostale visoke standarde programa, pa je zato i ne uključuje u paket.

Naravno, ono što stvarno stoji iza ova dva programa čini da je poređenje u smislu opreme i ostalih elemenata koji čine paket skoro neumesnim. *Tipotek* doživljava sudbinu za koju gotovo da je predodređen svaki domaći program. Na situaciju na našem tržištu koje svaki domaći napor potpuno obezvređuje, pridodati su i opšti teški uslovi u kojima se trenutno nalazimo, tako da je autor bio primoran da značajno spusti cenu. Niske cene, suženo tržište i opšta besparica osiromašili su opremu programa, koji se sada oslanja samo na *on line help* i *tutorial*.

Način distribucije i registracije je takođe promenjen, ali ova promena zaslužuje svaku pohvalu. Program sa sada slobodno može preuzeti sa *Sezama* i neograničeno deliti prijateljima, ali je za njegovu punu operativnost potrebno kupiti drajvere za pojedine formate fontova. Umesto ranijeg restriktivnog instaliranja sa diskete, sada program jednostavno raspakujete iz originalne arhive, pročitate identifikacioni broj kopije i dostavite ga autoru, nakon čega dobijate mali registracionu datoteku koja aktivira drajvere koje ste kupili. Cena drajvera za jedan format iznosi u proseku oko 20 DEM. To konkretno znači da pored 20 DEM za osnovnu registraciju treba platiti još oko 60 DEM, tako da za 80 DEM dobijate program koji radi sa *Adobe Type 1*, *TrueType* i *Corel Draw* formatima. Maksimalna cena je 100 DEM.

Iako će se mnogi obradovati ovako niskoj ceni, moram da izrazim žaljenje što je autor na nju primoran. Ona u krajnjoj liniji ne daje mnoge prostore za dalji razvoj niti za eventualni pokušaj izvoza programa, što je prava šteta. Iz razgovora koji sam u toku pripreme ovog teksta imao sa autorom programa, ipak sam stekao utisak da ga ovi tužni uslovi u kojima program živi neće pokolebati da nastavi sa daljim razvojem. Za dosadašnje i buduće korisnike ovog programa to je svakako dobra vest.

Kontakt adresa: Infotek, preduzeće za proizvodnju i prodaju informatičke opreme, usluge i inženjering, 11000 Beograd, Dunavski kej 5. Tel: 631-547, fax: 632-650

ZAŠTITA MEMORIJE

Posle seta instrukcija, dolazimo do najzanimljivije teme koja, zapravo, i predstavlja osnovni novitet mikroprocesora 80386/80486 u odnosu na prethodne *Intel*-ove generacije: bavićemo se selektorima, deskriptorima i zaštitom memorije na osnovu privilegija programa.

Dejan Ristanović

Konstruktori nekadašnjih kućnih računara nisu imali nikakve potrebe da brinu o zaštiti sisteme – program koji se izvršava je „vlasnik" svih komponenti računara i može da radi baš sve što mu odgovara, ako je rezultat neke od tih akcija totalno zamrzavanje sistema vlasnik će lako iščupati i ponovo ukopčati napajanje i sve će se vratiti u neko „nulto" stanje. Potreba za tim „vraćanjem u nulto stanje" uopšte nije bila tako retka kako bi se moglo pomisliti i u svakodnevnom radu je predstavljala više pravilo nego izuzetak – sećamo se da je po oglasima bilo toliko onih koji ugrađuju *reset* tastere na Spektrume da smo se prosto pitali kada će ulicom proći neka zaprega uzvikujući „staro gvožđe stare akumulatore kupujemo *reset* tastere na Spektrume ugrađujemo". Ni IBM PC nije bio ništa bolji – relativno često dešava se da se sistem „zaglupi", tako da ni čuveno *Alt Ctrl Del* ne funkcioniše konstruktori većine klonova su, srećom, predvideli *reset* taster na kućištu.

Nasilni prekid programa koji se „zaglupao" predstavlja malu (ili malo veću) neprijatnost ako je taj program bio jedini koji se izvršavao ali izuzetno ozbiljan problem ukoliko se u istom trenutku izvršava nekoliko programa što je ambicija autora svakog modernijeg operativnog sistema – zašto bismo na primer morali da resetujemo čitav računar i tako „ubijemo" sve programe koji se izvršavaju zbog jednog koji je „podivljao"? Treba osim toga obezbediti da svaki program ima pristup samo svojim podacima i onemogućiti ga da deluje na podatke drugih programa – *Core War* je lepa igra ali teško da bismo poželeli da se svakoga dana nekontrolisano odvija na našem PC-ju! Ovakvu zaštitu pokazuje se operativni sistem sam po sebi ne može da obezbedi – potrebno je da osnove te zaštite budu ugrađene u sam mikroprocesor. *Intel* je počeo da radi na tome dizajnirajući mikroprocesor 80286 – tu je implementirao takozvani zaštićeni režim rada (*protected mode*) ali kao „svet za sebe" nedovoljno povezan sa *real* režimom u kojem se izvršavaju svi DOS programi. Zato na AT računarima ni danas nema „pravog" višekorisničkog rada. Pri projektovanju 80386 stvari su dovedene od kraja što je rezultiralo dobrim višeprogramskim operativnim sistemima kao što su razne verzije *Unix*-a OS/2 2.0 i najavljeni *Windows NT*. Mikroprocesor 80486 je preuzeo čitav ovaj mehanizam sa 80386 – na tom planu nisu unesene nikakve značajnije izmene.

SELEKTORI, DESKRIPTORI I PRIVILEGIJE

Osnovni pojam na kome se čitava zaštita zasniva je **selektor**. Umesto da direktno adresira bilo koji objekat u sistemu program se poziva na njegov selektor na osnovu kojeg će mu operativni sistem odobriti (ili zabraniti) raspolaganje objektom. Selektor ukazuje na informaciju o objektu – tip adresu, veličinu prava pristupa i tome slično. Treba ipak znati da nijedan od ovih podataka nije upisan u sam selektor i to iz dva razloga. Pre svega „pamćenje" svih tih informacija bi učinilo selektor veoma obimnim pa bi ga bilo teško (tj. sporo) prosleđivati od programa do programa. Što je još važnije, loše (ili zlonamerno) napisani program bi mogao da modifikuje informacije upisane u selektor i tako pristupi objektu koji nije njegov. Zato je selektor zamišljen kao zapečaćena pošiljka koju „kuriri" nose od programa od programa ali ne mogu postupiti njenom sadržaju. Jedino sam operativni sistem može da pristupa sadržaju tog „koverta" koji nazivamo deskriptorom. Kao što mu ime i govori deskriptor sumira sve detalje o nekom objektu ti detalji za neki objekat mogu da se svedu na oznaku segmenta u koji je objekat upisan.





Slika 1. „Prstenasta" zaštita mikroprocesora 80486



Slika 2. Dopuštene i zabranjene komunikacije između prstenova



Slika 3. Pozivanje potprograma preko kapije

Deskriptori su grupisani u tabele deskriptora. Na osnovu selektora mikroprocesor „zna" koji deskriptor odgovara objektu, pa mu nije teško da pročita polje DPL (više o njemu kasnije) koje označava pravo pristupa segmentu. Na osnovu sadržaja tog polja, mikroprocesor donosi jednu od tri odluke. Pre svega, može da odobri pristup objektu što se u praksi najčešće dešava. Dalje može da odobri pristup objektu ali da nađe da je taj pristup trenutno nemoguć – program je na primer zatražio sadržaj neke memorijske ćelije koja je trenutno nedostupna jer je na način o kome smo govorili pre dva tri meseca odgovarajući segment prenesen na disk koji predstavlja virtuelnu memoriju. U tom slučaju program će imati priliku da preduzme odgovarajuće akcije i kada se steknu uslovi ponovi zahtev. Mikroprocesor, najzad može da zaključi da program nema pravo raspolaganja traženim objektom i o tome generišući odgovarajući interapt obaveštava operativni sistem. Ovaj će preduzeti odgovarajuće akcije koje se na „manjim" operativnim sistemima obično svode na suspendovanje ili nasilni prekid izvršavanja nekorektnog programa.

80486 deli sve programe na osnovu prava koja imaju u četiri grupe (slika 1). Najprivilegovaniji programi su na nivou 0 – to je zapravo samo jezgro operativnog sistema. Nivo 1 obično predstavlja operativni sistem, nivo 2 su korisnički programi koji mogu da pristupaju priključenom hardveru, a nivo 3 najmanje privilegovani programi korisnika. Videćemo da čitava ove organizacija toliko podseća na prstenove sa slike 1 da mnoge nezavisne publikacije ne govore o „nivoima privilegija" (što je *Intel*-ov termin) nego o prstenima kada se kaže da se program izvršava u nultom prstenu misli se zapravo na jezgro operativnog sistema čiji je nivo privilegija 0.

Konkretan operativni sistem ne mora da podržava sva četiri prstena. Sve verzije *Unix*-a na primer koriste samo nulti prsten za kernel i prsten 3 za korisničke programe. OS/2 koristi prstenove 0, 2 i 3 za operativni sistem, korisničke programe koji smeju da pristupaju periferijskim uređajima i ostale korisničke programe.

Nivo privilegija tekućeg programa određen je CS registrom i IOPL poljima EFLAGS registra. Samo se po sebi razume da program ne može da utiče na ova polja – u protivnom bi svako samome sebi mogao dodeliti najviše privilegije. Kada sa POPF ili POPFD pokušate da

izmenite ova polja izmene će biti ignorisana ukoliko se vaš program ne izvršava u nultom prstenu

Prema konvencijama mikroprocesora 80486 program može da pristupa samo podacima koji se nalaze na njegovom ili nekom od sledećih nivoa privilegije – program u prstenu 1 na primer, može da pristupa podacima sa prstena 1 2 ili 3 ali ne može da ispituje ili menja podatke iz nultog prstena. Što se tiče pozivanja potprograma ograničenja su još strožija – program može da pozove samo potprogram iz istog prstena što znači da ni najprivilegovaniji deo operativnog sistema ne može da pozove nejmanje privilegovani korisnički program iz trećeg prstena (slika 2)!

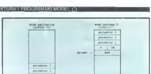
Ovakva ograničenja su čudna samo na prvi pogled – operativni sistem zbilja ne bi smeo da pod svojim privilegijama izvrši bilo koji korisnički program jer bi onda taj program mogao po želji da „vršlja" po sistemu. Sa druge strane korisnički program često mora da traži usluge operativnog sistema ali to ne bi smeo da radi nekontrolisano – ako bi mu se dopustilo da „skoči" na bilo koju adresu nultog prstena mogao bi pozvati neki potprogram koji nije predviđen za pozivanje (ili koji još gore nije predviđen da bude potprogram) i tako u većini slučajeva srušiti sistem a ako je vešto pisan i dobiti privilegije koje mu inače ne pripadaju. Zato su obezbeđeni „mostovi" za kontrolisane prelaske iz prstena u prsten – ti mostovi se u *Intel*ovoj terminologiji nazivaju kapijama (*gate*)

KAPIJE PROCESORA

Kapija je sistemski objekat (ima svoj deskriptor) koja ukazuje na proceduru u segmentu koda koja je nivoom privilegija odvojena od tog kodnog segmenta. Slika 3 pokazuje kako sve to deluje. Vidi se da je program iz četvrtog prstena preko susedne kapije dobio pravo da izvrši (ali samo da izvrši dakle ne i da menja!) potprogram iz trećeg prstena a ovaj je preko druge kapije pozvao potprogram iz prvog prstena. Tokom izvršavanja potprograma proces privremeno dobija potreban nivo privilegija kao da je za trenutak prešao u viši (ili niži) prsten čim se izvršavanje završi automatski se vraća raniji nivo. Mikroprocesor 80486 podržava četiri vrste kapija – preko *call* kapije „idu" standardni pozivi potprograma *interrupt* i *trap* kapije obezbeđuju obradu prekida dok je *task* kapija „najopštija" pošto se aktivira JMP CALL ili INT instrukcijom ili hardverskim interaptom

Pažljiviji čitalac će primetiti jedan ozbiljan problem čitavog mehanizma – s obzirom da se pri pozivu potprograma parametri (ako ne mogu svi stati u registre) smeštaju na stek oni se ne mogu smatrati sigurnim, a što je još gore ni sama adresa povratka više ne mora biti ono što je bila. Moglo bi se zamisliti da operativni sistem pozove neki korisnikov program, on promeni adresu povratka i tako ponovo „upadne" u neki deo nultog prstena koji nije predviđen za „upade" – tako ponovo dođosmo do problema zbog kojeg su kapije i uvedene! Slika 4 pokazuje kako se to rešava – deo steka se prilikom „prolaska kroz kapiju" kopira u privilegovani stek segment. Deskriptor svake kapije sadrži polje *count* koje sadrži broj 32-bitnih reči koje treba kopirati sa steka podešava se da taj broj obuhvati adresu povratka i onoliko parametara koliko pozvani potprogram najviše može da ima

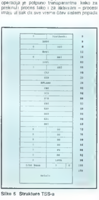
Svaka aplikacija dakle mora da ima onoliko stek segmenata koliko operativni sistem nivoa privilegija podrzava. Ovo zvuči kao prilično rasipanje ali na žalost, sve ima svoju cenu pa i sigurnosni mehanizmi – ako je prostor u RAM-u „tesan" uvek se može posegnuti za virtuelnom memorijom. Ako pouzdano znate da nikada nećete pozivati potprograme iz prvog prstena kreirajte deskriptor odgovarajućeg stek segmenta i obeležite ga kao *not present* – neće zauzimati nikakvu memoriju neće nikada biti pozvan i sve je u redu! Vredi još pomenuti da svaki segment steka nema svoj stek pointer – postoji samo jedan ESP. Pointeri na stek u ostalim prstenovima deo su sistemskog objekta koji se zove *task state segment* odnosno TSS



Slika 4 Promena stek segmenta pri prolazu kroz kapiju

TSS je poseban segment memorije koji procesor koristi za podršku višeprogramskog rada. Njegova struktura prikazana je na slici 5. Osim kopije svih registara i drugih vrednosti od značaja za docniji nastavak izvršavanja nekog procesa TSS sadrži i tri dodatna stek segmenta SS0 SS1 i SS2 i tri stek pointera ESP0 ESP1 i ESP2. Kada nastupi privremena promena privilegije programa zbog poziva preko kapije novi SS i ESP se formiraju iz odgovarajućeg polja TSS-a. *Task register* (TR) uvek ukazuje na trenutno aktivni TSS

Kada operativni sistem zahteva zamenu poslova svi registri aktivnog posla se smeštaju u njegov TSS, zatim se formira novi sadržaj TR-a i na osnovu njega registri pune parametrima iz TSS-a sledećeg procesa. Čitava ova operacija je potpuno transparentna kako za prekinuti proces tako i za aktivirani – procesi imaju utisak da sve vreme čitav sistem pripada njima i da ih niko ne prekida bar dok ne pokušaju neku akciju na koju nemaju pravo



Slika 5 Struktura TSS-a

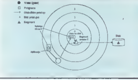
TABELE DESKRIPTORA

Kao što smo videli svaki objekat u sistemu (segment memorije TSS kapija, itd) ima svoj deskriptor. Deskriptori se grupišu u tabele kojih ima više – za dizajniranje operativnog sistema moraju postojati interapt tabela deskriptora (IDT) i globalna tabela deskriptora (GDT) dok je lokalna tabela deskriptora (LDT) opciona ali će svakako biti u upotrebi na svakom ozbiljnijem operativnom sistemu

Interapt tabela deskriptora (IDT) ukazuje na interapt vektore. Register IDTR sadrži apsolutnu **linearnu** adresu ove tabele i njenu maksimalnu dužinu – navođenje apsolutne adrese unekoliko odudara od uobičajenog segment offset adresiranja mikroprocesora 80486 ali je u datom slučaju neophodno da bi se ubrzala obrada prekida. Globalna tabela deskriptora (GDT) sadrži opise svih važnijih sistemskih objekata i održava je operativni sistem na njen početak i dužinu ukazuje registar GDTR. Lokalna tabela deskriptora kao što rekosmo nije neophodna ali obezbeđuje dodatni mehanizam zaštite koji je potreban ako se očekuje da će sistem biti napadnut ne od strane „podivljalog" programa nego nekog hakera koji želi da stekne privilegije koje normalno nema. Slika 6 ilustruje scenario takvog napada

„Napadač" je svestan toga da mu sigurnosni sistem mikroprocesora neće dopustiti da menja neku zonu memorije koja predstavlja deo samog operativnog sistema pa napada indirektno na kopiju operativnog sistema koja je trenutno na disku. U tom cilju poziva potprogram operativnog sistema koji piše na disk i prosleđuje mu adresu zone koju napada – to je recimo neki segment operativnog sistema koji je trenutno izbačen iz memorije i nalazi se na disku čekajući novi poziv. S obzirom da se potprogram operativnog sistema izvršava u nultom prstenu biće mu dopušten upis na disk i pri sledećem učitavanju neće biti unesen ispravan segment nego onaj koji je napadač modifikovao i preko kojeg će uz malo veštine postići sve što želi

Da bi se ovakvi napadi u startu onemogućili selektor je dobio polje RPL (*requested privilege level*) – slika 7 prikazuje kompletnu strukturu 16-bitnog selektora. U polje RPL svakog selektora treba upisati CPL (*current privilege level*) rutine koja ga poziva a zatim izvršiti instrukciju ARPL (*adjust requested privilage level*). Posle toga će sam mikroprocesor detektovati da je traženi nivo privilegija (RPL) selektora niži (tj numerički posmatrano veći) od DPL-a zahtevanog segmenta i odbiće da izvrši operaciju. Slika 8 prikazuje kako deluje taj mehanizam



Slika 6 Scenario „napada" na sigurnosni sistem 80486 ▲

Polje TI selektora identifikuje tabelu u kojoj se nalazi odgovarajući deskriptor. Nula označava deskriptor u globalnoj tabeli (GDT). Vrednost 43h upisana u polja selektora, na primer, označava osmi deskriptor u GDT-u (ako broj 43h predstavite binarno, dobiće se 01000011 pa će 11 biti RPL, 0 će biti TI a 01000 je dekadno 8). Prvi red GDT-a (GDT(0)) se nikada ne koristi tako da se vrednost selektora 0000h označava kao *null selector* (vrednost RPL kod *null* selektora se ignoriše, pa su 0, 1, 2 i 3 potpuno jednaki *null* selektori) i može se učitati u registar bez narušavanja mehanizma zaštite memorije. Na taj način se, zapravo, i kreiraju selektori novih objekata.

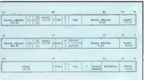
Ako je TI=1, selektor ukazuje na lokalnu tabelu deskriptora (LDT) koja je po svemu slična globalnoj, osim što se LDT(0) koristi ravnopravno sa ostalim deskriptorima. Lokalne tabele deskriptora se uglavnom kreiraju za svaki posao posebno, i to iz dva razloga. Prvi (i za razumevanje jednostavniji) razlog je u tome što je selektor 16-bitni, pri čemu indeksno polje ima svega 13 bita, što daje oko 8000 deskriptora. Kada bi to bilo sve, operativni sistem koji izvršava puno programa mogao bi da se nađe u nevolji kada „potroši" sve deskriptore – ovako, odvojena tabela za svaki posao naprosto znači gotovo neograničen broj deskriptora.

Drugi razlog je mnogo suptilniji i ponovo se svodi na zaštitne mehanizme. Slika 9 prikazuje šemu operativnog sistema koji sve deskriptore drži u GDT-u. Deskriptori 1-100 ukazuju na razne sistemske objekte iz nultog prstena, GDT(100) ukazuje na deskriptor kodnog segmenta programa A u trećem prstenu, GDT(101) na deskriptor segmenta podataka istog programa, dok GDT(102) i GDT(103) na sličan način ukazuju na program i podatke programa B. Jasno je da će bilo koji pokušaj programa A da utiče na tuđe podatke biti zaustavljen odgovarajućim interaptom (*protection fault*), ali se postavlja pitanje šta ako A pokuša da falsifikuje deskriptor? Šta ako kreira sasvim ispravan deskriptor koji ukazuje na tuđe podatke? „Napad" na neki od prvih 100 redova GDT-a će biti onemogućen, jer se radi o objektima iz prstena nula kojima program iz prstena 3 ne može nikako pristupiti, ali napad na GDT(103) ne bi bio onemogućen, pošto se i A i B nalaze u trećem prstenu, pa bi ispalo da jedan od njih, uz malo veštine, može da pristupi podacima koji su vlasništvo drugoga.

Rešenje ovog problema je u uvođenju lokalnih tabela deskriptora (LDT) za svaki posao. Sada će posao moći da falsifikuje samo redove u **sopstvenom** LDT-u, što nije opasno jer podaci i tako pripadaju tom programu; napad na GDT će biti zaustavljen, jer se radi o vlasništvu programa iz prstena 0. LDT, dakle, definiše virtuelni adresni prostor aplikacije, pri čemu svaki program ima svoj prostor koji ne može da se pomeša sa tuđima. Sam LDT je sistemski objekat koji ima deskriptor u GDT-u, ali je taj deskriptor „vlasništvo" operativnog sistema a ne korisnikovog programa!



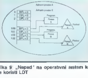
Slika 7 Format selektora



▶ Slika 10 Format 80486 deskriptora



Slika 8 Sprečavanje „napada" sa slike 6



Slika 9 „Napad" na operativni sistem koji ne koristi LDT

FORMAT DESKRIPTORA

Slika 10 prikazuje format sva tri tipa deskriptora koje 80486 podržava. Prvi je sistemski deskriptor koji opisuje LDT/TSS i druge slične objekte, drugi je deskriptor segmenta memorije, a treći deskriptor kapije. Sistemski deskriptori i deskriptori segmenta memorije su prilično slični (odvaja ih polje S koje ima vrednost 0 odnosno 1) i u osnovi ukazuju na istu stvar (i sistemski podaci su u memoriji!), ali postoji jedna veoma važna razlika – LDT/TSS deskriptor se **ne** može preneti u segmentni registar, tj. ne može se tretirati kao podatak koji se odgovarajućim instrukcijama čita ili menja. Razni sistemski programi (npr. *debugger*) koji imaju potrebu za ispitivanjem sadržaja ovakvog objekta moraju da kreiraju takozvani *alias* memorijski segment podataka koji ukazuje na istu zonu memorije, i onda toj memoriji pristupaju preko njega. Čak i za tako „obične" stvari kao što je samomodifikujući kod, moraju se kreirati *alias*, pošto 80486 ne dopušta upis podataka u segment koda. Samo se po sebi razume da za kreiranje *aliasa* preko kojeg će se pristupati tuđim podacima program mora biti visoko privilegovan.

Polje *Type* sa slike 10 precizno određuje vrstu deskriptora. Nula nije dopuštena (deskriptor je neispravan), 1 je TSS mikroprocesora 80286, 2 je LDT, 3 je zauzet 80286 TSS, 9 je 80386/80486 TSS, a 11 zauzet 80386/80486 TSS.

Deskriptor kapije se razlikuje od memorijskog i sistemskog deskriptora pošto nema potrebu za baznom adresom i limitom jer ne ukazuje na podatak u memoriji. No, on sadrži selektor **drugog** deskriptora, kao i ofset u okviru tog segmenta ako se radi o pozivu potprograma ili interapta. *Task gate* čuva selektor TSS-a dok se polje „deskriptor" ne koristi.

Broj 4 u polju *Type* označava 80286 kapiju, 5 je *task gate*, 6 je 80286 *interrupt gate*, 7 je 80286 *trap gate*, dok 12, 14 i 15 označavaju 80386/80486 *call*, *interrupt* i *trap gate* respektivno. Vrednosti 8, 10 i 13 su rezervisane i eventualno će se koristiti kod budućih *Intel*-ovih mikroprocesora.

Operativni sistemi koji se izvršavaju na 80286 (npr. OS/2 1.2) bez problema rade i na 80386/80486, što znači da ovi mikroprocesori potpuno podržavaju i deskriptore tipa 1, 3, 4, 6 i 7. „Pravi" 32-bitni operativni sistemi (npr. OS/2 2.1) koriste samo 32-bitne deskriptore – *Intel* napominje da korišćenje 16-bitnih sistemskih deskriptora (naročito TSS) na 32-bitnom operativnom sistemu može pod nekim okolnostima dovesti do nepredvidljivog ponašanja mikroprocesora.

U sledećim „Računarima" bavićemo se višeprogramskim radom (*multitasking*) na 80486.

PC386,486 VESA-LOCALBUS-32bit !!!

SC3040DX ISA	SC4033DX LOCALBUS	SC4050DX LOCALBUS
AMD 386DX-40MHz RAM 4Mb / 128Kb cache FLOPPY 1.2 Mb HD 105 Mb 13 ms ALPS 16bit IDE CONTROLER 16bit SVGA 1 Mb CIRRUS LOGIC - - WINDOWS AKCELER. 1280x1024 14" SVGA MONO MONITOR MINI TOWER, TASTATURA 101	**INTEL 486DX-33MHzVESA** RAM 4Mb / 256Kb cache FLOPPY 1.2 Mb + 1.44 Mb HD 250 Mb 16 ms TEAC 32bit VL BUS IDE CONTROLER 32bit VL BUS SVGA 1Mb CIRRUS - - WINDOWS AKCELER. 1280x1024 14" SVGA COLOR MONITOR MIDI TOWER, TASTATURA 101	**INTEL 486DX-50MHzVESA** RAM 4Mb / 256Kb cache FLOPPY 1.2 Mb + 1.44 Mb HD 250 Mb 16 ms TEAC 32bit VL BUS IDE CONTROLER 32bit VL BUS SVGA 1Mb CIRRUS - - WINDOWS AKCELER. 1280x1024 14" SVGA COLOR MONITOR MIDI TOWER, TASTATURA 101
1.800 DEM	3.450 DEM	3.800 DEM

386SX-33MHz,RAM2Mb,HD105Mb,FD1.2Mb,14"MONOSVGA,1Mb-WIN.AKC.......1550DEM

	CENE DELOVA	1-10 KOM.	11-30 KOM.
DISKOVI	HARD DISK 105Mb ALPS 13 ms	500 DEM	450 DEM
	HARD DISK 170Mb CONNER 17 ms	650 DEM	600 DEM
	HARD DISK 250Mb TEAC 16 ms	850 DEM	750 DEM
	HARD DISK 540Mb CONNER 12 ms IDE III SCSI	1950 DEM	1750 DEM
	FLOPPY 1.2Mb / FLOPPY 1.44Mb	140 / 120 DEM	130 / 110 DEM
KARTICE	16 bit 2HD, 2FD CONTROLER + 2RS232 + 1 PARALEL	60 DEM	40 DEM
	32 bit VESA LOCAL BUS 2HD, 2FD CONTROLER / SA 4Mb CACHE	150 / 900 DEM	130 / 500 DEM
	SCSI ISA 2HD, 2FD CONTROLER 4Mb CACHE	1100 DEM	1000 DEM
	SCSI VESA, EISA 2HD, 2FD CONTROLER 4Mb CACHE	1200 DEM	1100 DEM
	ETHERNET 16 bit NE 2000	190 DEM	160 DEM
	FAX - MODEM KARTICA MNP5, V.42 BIS	220 DEM	160 DEM
SVGA	MONO MONITOR 14' 1024x768	250 DEM	220 DEM
	COLOR MONITOR 14' 1024x768	650 DEM	600 DEM
	COLOR MONITOR 15' 1280x1024 RAVAN EKRAN NON INT	990 DEM	900 DEM
	COLOR MONITOR 17' 1280x1024 RAVAN EKRAN N.I. LOW RAD.	2200 DEM	1990 DEM
	CIRRUS LOGIC 1Mb, 1280x1024 WINDOWS AKCELERATOR	200 DEM	160 DEM
	32 bit VESA L. BUS CIRRUS LOGIC 1Mb, 1280x1024 WIN AKCEL.	260 DEM	220 DEM
	32 bit VESA L. BUS S3 2Mb, 1280x1024 WINDOWS AKCELERATOR	450 DEM	400 DEM
PLOČE	386 SX-33MHz	220 DEM	200 DEM
	386 DX-40MHz-128Kb CACHE	350 DEM	300 DEM
	VESA L. BUS 486 DX-33MHz-256Kb CACHE	1000 DEM	950 DEM
	VESA L. BUS 486 DX-50MHz-256Kb CACHE	1350 DEM	1250 DEM
	VESA L. BUS 486 DX2-66MHz-256Kb CACHE	1650 DEM	1450 DEM
PRINTERI	LASER HP IV-2Mb 600x600 / RAM 4 Mb	3650 / 450 DEM	3600 / 400 DEM
	EPSON A4, LQ 570-24 PIN, KABL (YU SET 60 DEM)	900 DEM	650 DEM
	EPSON A3, FX 1050-9 PIN, KABL, YU SET	1150 DEM	1050 DEM
	EPSON A3, LQ 1070-24 PIN (YU SET 50 DEM)	1260 DEM	1150 DEM
	EPSON A3, LQ 1170-24 PIN (YU SET 50 DEM)	1400 DEM	1300 DEM
	PANASONIC A4, 1121I-24 PIN, KABL !!!!!	600 DEM	660 DEM
RAZNO	MEMORIJE SIMM 1Mb / SIMM 4Mb	50 / 350 DEM	70 / 300 DEM
	MATEMATIČKI KOPROCESOR IIT 387DX-40MHz	200 DEM	160 DEM
	MINI TOWER + 200W / MIDI TOWER + 200W	160 / 200 DEM	140 / 160 DEM
	PROFESIONALNA TASTATURA 101 / ALPS SA KLIKOM	70 / 120 DEM	60 / 110 DEM
	MIŠ - MICROSOFT KOMPATIBILAN, SA PODMETAČEM	60 DEM	40 DEM
	STREAMER 120Mb / 250Mb	550 / 750 DEM	500 / 650 DEM

ISPORUKA ODMAH **GARANCIJA 12 MESECI**

ŽIRO RAČUN: 60815-601-64787 | RADNO VREME 9h-17h

TEL: 011/332-607
FAX: 011/345-126

PREDUZEĆE
ŠUTLIĆ
&
MIKRO DIZAJN

11000 BEOGRAD
Kosovska 32, I sprat
JUN 1993.

MEMORIJA I MODELI

Tekst namenjen onima koji sa tek bore sa pokazivačima, memorijskim modelima, stekom, heap-om, ali i onima koji bi hteli da razjasne neke stvari. Recimo, šta je misteriozna greška *Null Pointer Assignment*? Da li je maksimalan niz veličine 65536 ili možda manje? U čemu je razlika između *malloc* i *allocmem*? Počnimo sa kratkim uvodom o seriji 86 procesora...

Bojan Petrović

Intelov 8086 mikroprocesor ima segmentiranu arhitekturu memorije. Mada ima ukupni adresni prostor od 1Mb, projektovan je tako da u jednom trenutku može adresirati maksimalnih 64k memorije. Ovaj komad se popularno naziva segmentom, odakle potiče i upravo upotrebljni naziv „segmentirana arhitektura".

Procesor 8086 posebno pazi na četiri važna segmenta: **code, data, stack** i **extra** (po naški: segment za kod, podatke, stek i posebari segment). Prvi, logično, skladišti instrukcije programa, drugi podatke, treći stek, a poslednji se koristi za neke posebne, za sada nedefinisane podatke. Segmenti se adresiraju posebnim registrima procesora, nazvanim segmentnim registrima CS, DS, SS i ES. Svaki od njih je 16-bitni, a nazivi su zapravo skraćenice imena navedenih segmenata, respektivno. Tu su takođe i registri opšte namene, koji su u većini slučajeva slobodni za korišćenje po želji programera, ali u nekim slučajevima imaju određenu namenu: AX se mora koristiti u nekim matematičkim operacijama, BX se koristi kao indeksni registar, CX za LOOP i string instrukcije, a DX se takođe koristi za neke matematičke instrukcije. Zatim, tu su registri specijalne namene koji su veoma „opasni" jer imaju tačno definisanu namenu, veoma važnu za funkcionisanje sistema. SI i DI su indeksni registri, mogu se koristiti za neke naše namene, a Borland ih koristi za registarske varijable. SP registar je takozvani „stek pointer", a pokazuje na vrh steka. BP je sekundarni stek pointer, a pokazuje na mesto na steku gde su smešteni podaci koji su trenutno potrebni funkciji u toku. Ovaj poslednji je naročito zgodan za kvarove, jer njegovom promenom program gubi znanje o položaju stvarnih parametara date funkcije (recimo), a rezultat je potpuno pogrešan rad programa (ili pad sistema). Naime, parametri se adresiraju relativno prema ovom registru, a on sam pokazuje na položaj BP iz prethodne seanse stavljanja podataka na stek (recimo, funkcije iz koje je ova trenutna pozvana).

Adresa sa kojom operišemo se sastoji iz dva dela: segmenta i ofseta. Ako pretpostavimo da posmatramo **data** segment (registar je DS) i da se nalazimo na segmentu 2F84h (heksadecimalno), a zanima nas stvarna adresa 0352h lokacije odatle. Adresu bi dakle pisali kao 2F84:0352h, dok bi stvarna adresa morala da se preračuna: pomeramo segmentni deo za jedno heksadecimalno mesto ulevo, dopunjavamo nulom i sabiramo sa ofsetnim delom, odnosno

```
 2F840
+ 0352
------
 2FB92h
```

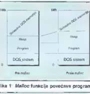
i me dobijamo apsolutnu adresu 2FB92h. Odavde odmah vidimo da adresa prikazana u paru segment-ofset nije jednoznačna, jer recimo adrese 0000:0123h, 0002:0103h i 0012:0003h su sve tri zapravo iste apsolutne adrese. Takođe, može se desiti i da segmentni registri pokazuju na isti segment, što znači da naš program u istih 64K more da smesti sve pomenute segmente.

Na kraju, jedna konvencija: dogovorićemo se da će za nas memorija rasti ka višoj adresi što ćemo zvati „gore". Ekvivalentno, kada kažemo „dole", govorimo o nižoj adresi.

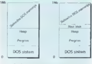
POKAZIVAČI

Pokazivač je memorijska lokacija koja čuva memorijsku adresu nekog objekta kao svoj sadržaj. Kada izvršite deklaraciju pokazivača kompajler će odvojiti odgovarajući prostor u memoriji potreban da se u njega smesti neka adresa. Ukoliko je pokazivač deklarisan globalno, adresa će automatski biti postavljena na 0000 u slučaju **near**, ili 0000:0000 u slučaju **far** pokazivača. Ove dve vrednosti zbog svog značaja imaju i ime NULL. Naime, uobičajen postupak je da se pre prve prave inicijalizacije svi pokazivači postave na NULL iz više razloga, jedan od njih je svakako lakše pronalaženje grešaka u programu. U svakom slučaju pre dodeljivanja neke smislene adrese pokazivaču, nazivaćemo ga neinicijalizovanim.

Inicijalizacija obuhvata dodeljivanje adrese nekog objekta datom pokazivaču, što se može učiniti odvajanjem dinamičke (**heap**) memorije pomoću **malloc, farmalloc, calloc, farcalloc, realloc, farrealloc, allocmem** ili dodeljivanjem neke poznate adrese (recimo neke druge varijable). Kada smo jednom inicijalizovali pokazivač, on je spreman za dalji rad.



Slika 1: ***Malloc* funkcija povećava program**



Slika 2: ***Allocmem* funkcija proizvodi novi blok memorije**

Prethodni suvoparni tekst ćemo bolje shvatiti kroz jednostavan primer. Poigrajmo se sa jednim pokazivačem:

int *p; Obična deklaracija jednog pokazivača: deklarisali smo ga kao pokazivač na varijablu tipa **int**.

***p=123;** NE! Pokazivač **p** sada sadrži neku nama nepoznatu adresu na koju mi pokušavamo da smestimo 123. Šta ako **p** pokazuje na neku sistemsku promenljivu? Dakle, još nismo inicijalizovali **p**.

int i; Obična deklaracija jedne promenljive tipa **int**: kompajler automatski odvaja potreban prostor za nju.

p=&i; Pokazivaču **p** dodeljujemo adresu promenljive **i**, čime vršimo njegovu inicijalizaciju: **p** dakle sadrži adresu prostora odvojenog za **i**, odnosno pokazuje na **i**.

p=i; Pokazivaču **p** sada dajemo vrednost **i**, odnosno on od sada pokazuje na lokaciju čija adresa je bila smeštena u **i**.

p=(int *)malloc(sizeof int); Pokazivaču **p** sada dodeljujemo neki prostor stvoren komandom **malloc**; pokazivač je ovim inicijalizovan, ali pokazuje na prostor koji ne sadrži ništa pametno.

***p=123;** **p** sada pokazuje na prostor koji čuva vrednost 123.

p=123; Menjamo sadržaj samog pokazivača, odnosno teramo ga da pokazuje na lokaciju 123: opet ne znamo na šta on to pokazuje, odnosno pokazuje na ono što se slučajno zateklo na lokaciji 123.

***p=123;** Pokazivač **p** sada pokazuje na lokaciju 123, u kojoj se nalazi zapisan broj 123.

Postoje tri tipa pokazivača: **near, far** i **huge** (u slobodnom prevodu: bliski, daleki i ogromni).

Near pokazivač ima veličinu od 16 bita, i sadrži samo ofset adresu neke lokacije, dok segment adresu saznaje iz odgovarajućeg segmentnog registra. Takva adresa očigledno ima raspon od 0 do 64K. Nad njim se mogu vršiti neke standardne operacije, kao što su inkrementiranje i dekrementiranje, pa i sabiranje i oduzimanje. Međutim, pokušaj inkrementiranja pokazivača koji sadrži FFFFh će rezultirati „premotavanjem" na 0000h, a odgovarajuće suprotan efekat bi se dogodio pri dekrementiranju pokazivača koji sadrži 0000h. Takođe, **near** pokazivači se mogu međusobno i porediti standardnim relacionim operatorima, bez ikakvih problema, jer se uvek nalazimo u istom segmentu.

Far pokazivači imaju veličinu od 32 bita, što znači da mogu da sadrže kompletnu adre-

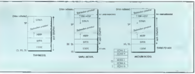
su u paru segment-ofset. Opseg ovakvog pokazivača se kreće od 0 do 1Mb. Bilo koja operacija dodavanja ili oduzimanja nad ovom vrstom pokazivača utiče samo na ofset deo adrese, što znači da će ovim operacijama biti moguće pomerati pokazivač samo u „prozoru" od 64Kb. Ovo nas odmah vraća na malopre pomenuti slučaj: ukoliko inkrementiramo 1234:FFFFh dobićemo 1234:0000h i obratno. Interesantna i važna činjenica vezana za **far** pokazivače je sledeća već pomenuta osobina: moguće je imati više različitih **far** adresa koje zapravo pokazuju na istu lokaciju. 1234:0020h i 1230:0060h predstavljaju apsolutnu adresu 12360h! Sada već naslućujemo da poređenje **far** pokazivača zapravo poredi samo ofset delove, što znači da bi poređenje ab gde je a=1234:0020 a b=1230:0060 bilo uspešno, mada su, kao što rekosmo, ovo iste adrese. Operacije jednakosti i nejednakosti („==" i „!=") koriste pri radu svih 32 bita, tako da je poređenje sa NULL moguće i poželjno iz već navedenih razloga.

Huge pokazivači su veličine 32 bita. Kao i prethodni, čuvaju adresu u paru segment-ofset i adresiraju 1Mb memorije. Mogu se inkrementirati, dekrementirati ali i porediti. Naime, kompajler normalizuje ovu vrstu pokazivača, što zapravo znači da će se što veći deo adrese „smestiti" u segment deo, a ofset deo će biti maksimalno 15. Ovaj broj je sasvim logičan, jer 16 predstavlja granicu jednog segmenta (tačnije, 16 apsolutna adresa je 0001:0000h adresa, nenormalizovan pokazivač sa ovom apsolutnom adresom bi sadržao 0000:0016). Poređenje dakle može funkcionisati jer sada imamo jednu i samo jednu adresu za datu lokaciju. Logično, gore opisana situacija sa „premotavanjem" ofset dela pri inkrementiranju ovde se ne može desiti. Međutim, možda najbitnija stvar ovde je da **huge** pokazivači nikad nisu podrazumevani, već se to mora naglasiti (**void huge*p**). **Huge** pokazivači su se pojavili iz sledećih, već rečenih razloga: jednoznačnost adrese, mogućnost poređenja i tačno inkrementiranje i dekrementiranje. Konačno, zbog stalnog normalizovanja, rad sa **huge** pokazivačima je daleko sporiji nego sa **far** verzijom, tako da Borland uvodi i brže **huge** pokazivače, gde se normalizovanje vrši samo u blizini granice segmenta.

MODELI

I opet se srećemo sa čudnim nazivom. Naravno, pogađate, nije reč o gospođicama mačijeg hoda, nego o memorijskim modelima, koji između ostalog, određuju veličinu **data** i **code** segmenata, broja i vrste dinamičkih memorija, vrstu pokazivača i tako dalje. U osnovi možemo nabrojati (opet na izvornom engleskom jeziku) sledeće tipove memorijskih modela: **tiny, small, medium, compact, large** i **huge** (ili po naški: malecki, mali, srednji, kompaktni, veliki i ogromni). Prva tri, na osnovu prethodnog poglavlja, možemo nazvati i **near** modelima, jer sva tri imaju **near data** pokazivače, dok preostala tri imaju **far data** pokazivača, a oni su za nas obično od najvećeg interesa (a moglo je i po osobini da prve tri imaju i **near** i **far heap**, a poslednja tri samo **far**). Takođe, prve tri moraju u istom segmentu (64K) smestiti **near heap**, **data** segment i stek (a **tiny** model i **code** segment), dok preostali mogu imati posebne segmente za stek, višestruke **code** i **data** segmente i tako to. Sve ovo će biti mnogo jasnije posle pregleda tabele 1.

Tiny model je najmanji od svih, gde sve četiri segmentna registra pokazuju na isti segment, tako da je programu dato samo 64K za kod, podatke i stek. Uvek se koriste **near** po-



Slika 3 Strukture *near* memorijskih modela

kazivači. Zgodan je za veoma male programe.

Small ima različite **code** i **data** segmente, tako da program raspolaže sa 64K za kod i 64K za podatke i stek. I ovde se uvek koriste **near** pokazivači. Zgodan je za većinu normalnih programa srednje veličine.

Medium model, za razliku od prethodnog, ima **far** pokazivače za kod, što znači da podaci i stek moraju zauzeti maksimalno 64k, a program se može protezati na celih 1Mb. Odgovara velikim programima sa veoma malo podataka u memoriji.

Compact model je inverzan prethodnom: **far** pokazivači se koriste za podatke, a ne za kod. Ovde kod i stek ostaju ograničeni na 64k, a podaci se protežu na 1Mb. Odgovara malim programima sa jako mnogo podataka u memoriji.

Large model koristi **far** pokazivače i za podatke i za kod, dajući svakome po 1Mb, dok stek i dalje ostaje na svojih 64k. Ovo je istovremeno i najčešće korišćeni model za ozbiljnije programe. Mada po osobinama odgovara velikim programima sa mnogo podataka, u mnogo slučajeva su potrebni **far** pokazivači, pa se stoga prilika koristi ovaj model.

Huge model takođe ima sve pokazivače tipa **far**, uz to da stek sada može biti i veći od 64K.

Interesantna posledica segmentirane arhitekture je da pri prevođenju nekog programa rezultujući kod uvek mora stati u jedan segment! To znači da je ponekad potrebno program pre prevođenja iseckati na module, gde će svaki biti preveden za sebe, a zatim ih linkovati u izvršnu verziju. Ovo, da naglasimo, važi i za **large** i **huge** modele.

Borlandov C(++) nudi još neke egzotične stvari u vezi sa modelima i pokazivačima. Standardnom ANSI C-u dodaju se modifikatori „near", „far", „huge", „_cs", „_ds", „_es", „_ss" i „_seg", koji se koriste za modifikovanje pokazivača i funkcija, uz neka ograničenja. Prva tri mogu da modifikuju podrazumevani tip pokazivača za dati model, što znači da ne samo da možemo imati **near** pokazivače u **far** modelima, nego i **near** funkcije. Recimo, ako u nekom **far** modelu koristimo rekurzivnu funkciju, stalno pozivanje nje same sa podrazumevanim **far** tipom će puniti stek sa nepotrebno mnogo podataka i trošiti previše vremena. Modifikovanje funkcije na **near** (što je u redu, jer funkcija zove samu sebe, a sigurno se dakle nalazi u istom segmentu) će umanjiti potrošnju steka i ubrzati njen rad.

```
double pow(double x, int exp)
{
    if (exp<=0)
        return(1);
    else
        return(x*pow(x,exp-1));
}
```

postaje

double near pow(double x, int exp)

što, da ponovimo, omogućava pozivanje ove funkcije samo iz njenog segmenta. Ovakve funkcije moraju biti definisane ili deklarisane pre prvog pozivanja, da bi kompajler znao koji poziv da upotrebi. Pametna praksa u ovim slučajevima je i istovremeno deklarisanje ovakvih funkcija statičkim, kako se njihovo ime ne bi pojavljivalo van datog modula (a podsetimo se, kod svakog modula, čak i u **far** modelima, ulazi u svoj segment).

Iz potpuno istih razloga se može modifikovati tip pokazivača: u **far** modelu modifikujemo pokazivač na **near** da bismo izbegli nepotrebnu potrošnju steka i vremena, u **near** modelu modifikujemo pokazivač na **far** tip da bismo dosegli nešto van datog segmenta. Međutim, ovde mogu nastati i interesantni problemi. Ako funkcija, recimo, očekuje kao parametar **far** pokazivač, a damo joj modifikovani **near** pokazivač, ne obavestivši prethodno prototipom kompajler o tome, na stek će otići 16 bita **near** pokazivača i još 16 nedefinisanih bitova. Rezultat će biti verovatno neočekivan. Ukoliko smo, dakle, prototipom obavestili kompajler o ovome, on će pri preuzimanju **near** pokazivača umesto nepoznatih 16 bita za segment na stek staviti sadržaj DS registra.

Sledeće četiri nabrojana modifikatora proizvode specijalne **near** pokazivače:

char _ss *p;

će proizvesti **near** pokazivač koji pokazuje negde na stek segment, a sadrži samo odgovarajući ofset. Poslednji modifikator rezultuje **near** pokazivačem za neki segment, odnosno 16-bitnim pokazivačem koji uvek predstavlja segment adresu:

char _seg *p;

gde se uvek podrazumeva ofset od 0000h. Za ovaj modifikator je vezan i niz ograničenja: nije moguće korišćenje ++, --, += i -= operatora nad njim, ne mogu se dva ovakva pokazivača oduzimati, sabiranje segmentnog i **near** pokazivača rezultira **far** kombinacijom (segmentni daje segment, a **near** ofset) i tako dalje.

HEAP

je poseban prostor odvojen za korisnikove dinamičke podatke, i zato se popularno naziva dinamičkom memorijom. Tu se, uobičajeno, smeštaju svi oni podaci koje korisnik može uneti, a čiju veličinu, broj ili tip nismo u stanju da odredimo pre prevođenja programa. Od neke maksimalne količine memorije dodeljene **heap**-u, potreban prostor za naše potrebe odvajamo već pomenutim **malloc**, **farmalloc**, **calloc, farcalloc, realloc** ili **farrealloc** naredbama, a po prestanku potrebe za sačuvanim podacima, uklanjamo ih sa **heap**-a naredbama

predviđenim za to (čime, dakle, oslobađamo zauzeti deo memorije).

Razlikujemo dve vrste: **near** i **far**. **Near heap**, ako postoji, počinje odmah posle podataka i raste ka steku, koji je smešten na kraju datog segmenta (slika 3). Slobodni međuprostor mogu koristiti kako koma treba – **heap** raste nagore, a stek nadole. Maksimalna veličina prostora za ova dva se može unapred odrediti globalnim promenljivim **_heaplen** i **_stklen**, što znači da se može desiti da neki od njih (ili oba) ne dobije toliko jer mu je onaj drugi već zauzeo potreban prostor. Klasična greška koja odavde proizilazi je odvajanje, recimo, pravilke količine memorije za **heap**, čime se zadire u oblast potrebnu steku. Ovo će, u suštini, proizvesti samo upozorenje (možda i grešku) pri prevođenju, ali tokom rada programa može odvesti do potpunog poremećaja memorije i pada sistema. Ovo se zato izbegava proverom stvarno dostupne količine **heap** memorije korišćenjem **coreleft** pre svake alokacije.

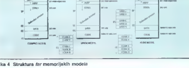
Far heap počinje od steka (iznad njega, slika 4) i raste nagore, ka granici od 1Mb. Početna veličina **far heap**-a je 0 bajtova. Kada program pokuša sa alokacijom memorije, poziva se DOS da veličinu programa poveća za zatraženu količinu bajtova (a to se radi u komadima od 16 bajtova). Očigledno, pokazivači koji se ovde koriste moraju biti bar **far**, jer govorimo o različitim i višestrukim segmentima. Ovde je važno reći da trenutno aktivni programi, a ne samo naš, potpuno ravnopravno traže od DOS-a dodatnu memoriju, što znači da sada ne možemo u **far** modelima korišćenjem **_heaplen** varijable rezervisati neku količinu **heap** memorije.

Ovde se postavlja problem dve vrlo slične stvari. Komanda **malloc** odvaja potrebnu količinu memorije sa **heap**-a i stavlja je nama na raspolaganje, dajući nam adresu tog komada. U **far** modelima, kao što smo rekli, memorija će biti zapravo zatražena od DOS-a, a veličina programa uvećana za toliko. Sa druge strane, **allocmem** takođe od DOS-a traži neku količinu memorije i vraća nam adresu. Gde je razlika?

Kada jedan program krene sa radom, DOS mu daje blok memorije i pazi da neki sledeći ili prethodni programi ne zadiru u taj blok. Ako je potrebno nešto memorije na **far heap**-u, taj blok se širi nagore (rasteže se), pri čemu DOS i dalje zna gde su njegove granice (slika 1). Ako sada pozovemo **allocmem**, DOS neće smatrati da ta količina memorije pripada programskom bloku, već da je reč o alokaciji sledećeg bloka memorije pod DOS-om. Po dogovoru (tačnije po podrazumevanom metodu alokacije), ovaj novi blok će biti smešten odmah posle progrmskog (slika 2). Kao rezultat ovog, program više neće moći da povećava svoju dinamičku memoriju, jer, rekli smo, **far heap** raste nagore od steka, a tu se sad nalazi sledeći blok. Kao rezultat, alokacija **heap** memorije neće uspeti (funkcija će vratiti NULL) i naš program neće moći dalje.

Naravno, ovom problemu ima rešenja. Ako baš moramo da koristimo obe vrste alokacije, pre korišćenja **allocmem** ćemo sistemu saopštiti da novi blok memorije ne smešta odmah posle trenutnog („first fit" metod), nego od vrha memorije nadole („last fit" metod). Ovo se radi o standardnim DOS pozivima, gde uvek posle promene podrazumevanog metoda na neki drugi moramo vratiti stanje na staro:

```
_AX=0x5801;
_BX=2; /* last fit */
geninterrupt(0x21);
/* ovde je alokacija */
_AX=0x5801;
_BX=0; /* povratak na first fit obavezno */
geninterrupt(0x21);
```



Slika 4 Struktura *far* memorijskih modela

Memorijski modeli

	TINY	SMALL	MEDIUM	COMPACT	LARGE	HUGE
Near Heap	Da	Da	Da	Ne	Ne	Ne
Far Heap	Da	Da	Da	Da	Da	Da
Code pokazivači	near	near	far	near	far	far
Data pokazivači	near	near	near	far	far	far
Posebni stack segment	Ne	Ne	Ne	ea	Da	Da
Višestruki code segmenti	Ne	Ne	Da	Ne	Da	Da
Višestruki data segmenti	Ne	Ne	Ne	Ne	Ne	Da

Osobine memorijskih modela

STEK

Stek je prostor posebno odvojen za podatke koji su trenutne prirode: parametri funkcija, automatske varijable, povratni rezultati, a pre svega **floating point** emulator. On uvek zauzima prvih 416 bajtova steka, i to od SS:0000h do SS:0415h, gde je SS stek segment (naravno, sve ovo ukoliko je dozvoljena emulacija). Za svaku funkciju koja se poziva, na stek se smeštaju redom svi njeni parametri, što bi rekli sa desna na levo, a za njima povratna adresa. Zatim slede sve automatske varijable koje se nalaze u telu funkcije i rekurzivno svi dalji pozivi funkcijama iz date funkcije. Pri izlasku iz funkcije sve automatske varijable se brišu sa steka, uzima se povratna adresa i eventualno ostavlja povratni podatak.

Kao što smo već rekli, stek i **heap** se u **near** modelima bore za prostor ravnopravno. Ukoliko smo dozvolili proveru ispravnosti steka, kompajler će posebnim kodom u **near** modelima upoređivati položaj vrha **heap**-a i vrha steka (stek, da ponovimo, raste ka nižim memorijskim lokacijama). Ukoliko između ove dve tačke nema mesta za potrebni podatak, generiše se greška. U **far** modelima stek je u sopstvenom segmentu, tako da se sada upoređuje vrh steka sa veličinom odvojenom za njega pomoću **_stklen**. Ove provere nisu bezgrešne, tim pre što se u mnogim komercijalnim bibliotekama pri prevođenju isključila provera steka. Takođe, kao što ćemo dodatno objasniti u sledećem poglavlju, ako pomoću **_stklen** damo steku maksimalnu moguću veličinu, može se desiti, baš kao kod **near** pokazivača, da se SP kada dođe na sam kraj steka premota na SP:0000 i počne da prepisuje onih 416 bajtova **floating point** emulatora. Ovo pak neće obavezno srušiti program, ali će rezultati računanja sa pokretnim zarezom polako gubiti smisao.

NEKE KLASIČNE GREŠKE

Jedna od najklasičnijih grešaka u radu sa pokazivačima i modelima je korišćenje neinicijalizovanih pokazivača. Ako otvorimo jedan automatski pokazivač (znači u telu funkcije), za njega će se prostor odvojiti na steku. U tom trenutku njegov sadržaj će biti jednak upravo onoj vrednosti koja se u tom trenutku našla na steku, i može se dogoditi da program radi na prvi pogled bez problema, ako ta slučajna adresa pokazuje na neki bezopasan prostor. Radi otklanjanja ovakvih grešaka, da ponovimo, dobra praksa je „inicijalizovanje" pokazivača na NULL, tako da uvek možemo ustanoviti da li je on samo dobio neku vrednost koja je pogrešna ili je stvarno neinicijalizovan.

Rad sa pokazivačima bez korišćenja ALLOC.H je još jedna sjajna greška. Bez njega, kompajler će za sve funkcije koje su inače u njemu definisane smatrati da vraćaju celobrojne vrednosti. U **near** modelima ovo i nije veći problem, jer su **near** pokazivači po veličini baš 16 bita, ali u **far** modelima će pola pokazivača (odnosno adrese) biti izgubljeno, i to baš segmentni deo. Naravno, opet se može desiti da program radi, ali nikad nećemo znati do kada.

Sve funkcije koje alociraju memoriju imaju običaj da u slučaju greške vrate NULL pokazivač. Zaboravljeno proveravanje ove mogućnosti je još jedna tipična greška. Već smo rekli, NULL pokazivač u suštini predstavlja adresu 0000:0000h, mada se često tumači i kao greška ili neinicijalizovani pokazivač. Ukoliko neka funkcija vrati NULL kao rezultat nemogućnosti alokacije memorije, pokvarenog niza memorijskih blokova ili bilo čega drugog, pokušaj korišćenja dobijene vrednosti će nas dovesti do „ubadanja" po listi pokazivača na sistemske prekide, što redovno obara sistem.

Oslobađanje dinamičke memorije funkcijama **free** i **farfree** ima smisla samo ako smo tu memoriju stvarno alocirali. Pokušaj oslobađanja memorije koja nije poreklom iz neke od pomenutih **malloc**, **calloc** i sličnih funkcija može dovesti do velikih problema.

Indeksiranje van dozvoljenih granica se može dogoditi i sa automatskim i sa dinamičkim nizovnim varijablama. Niz veličine 50 ima opseg indeksa od 0 do 49, što znači da **niz[50]** ne pripada njemu. Smeštanje neke vrednosti u ovu lokaciju će dovesti do nepredvidljivih rezultata, jer ćemo verovatno vrednost upisati

Nastavlja se na strani 83

DEJAN NIJE KOD KUĆE...

Kako je objektno programiranje u „Računarima" prisutno već skoro godinu dana, čini se da je vreme da se razotkrije šta je iza kulisa. Ovo je retka prilika da gledate majstora na delu.

Dejan trenutno nije kod kuće. Ostavite poruku

Ćao, Marko ovde. Čuj, raspitao sam se za onu analogiju između programskih i prirodnih jezika i taj tvoj tip je potpuno u pravu. Ispada da je većina zapadnjačkih jezika proceduralno orijentisana, što će reći da su organizovani oko radnji! Na drugu stranu, kineski je potpuno objektno-orijentisan jer organizovan je oko imenica. U stvari, ne znam koliko je objektno orijentisan jer ne znam za nasleđivanje, ali bar podržava apstraktne klase. Nekako sam znao da su ti objekti u stvari komunistička zavera. Čujemo se!

Biip

Dejan trenutno nije kod kuće. Ostavite poruku

Voja ovde. Kad ćeš već jednom da me pustiš da gostujem u „Računarima"? Znaš ono: „Upravo sam dobio *mail* od mog druga Voje i on kaže bla bla bla bla..." Čisto da impresioniram devojke. Nevezano za to, mogao bi da malo pišeš o razlikama između nasleđivanja i subtipizacije. Naime, tipizacija je, po Buču, čisto način da se onemogući korišćenje jednog objekta namesto drugog, ili da se to omogući u veoma ograničenoj količini. Na drugu stranu, nasleđivanje je odnos između klase u kome jedna klasa nasleđuje izgled i ponašanje drugih klasa. Sav ovaj mambo-džambo zapravo znači da nasleđivanje ne mora da da „jeste" odnos, tj. nasleđena klase ne mora da ima isti interfejs kao i roditeljska klase. To što su u većini jezika ta dva pojma veoma bliska samo pokazuje da je to tako lakše realizovati, ali ne i da je bolje. I još jedna stvar. Čini se da nijedna OOA/OOD metodologija ne podržava u pravom smislu reči *reuse*, višestruko korišćenje. Svi oni to ostavljaju programeru na dušu i time praktično peru ruke od detalja implementacije. Nisam siguran da je to dobro?

Biip

Dejan trenutno nije kod kuće. Ostavite poruku

Ej. Opet Voja. Malo sam razmišljao i pazi sad. Recimo da hoćemo da napravimo dve klase – **list** i **dlist** – koje bi predstavljale uvezanu i duplo uvezanu listu. Da li tu treba da klasa **dlist** nasleđuje klasu **list** jer je lista po kojoj možeš da se krećeš u oba smera zapravo samo specijalizacija liste po kojoj možeš da se krećeš u jednom smeru? Mislim, **dlist** može sve što može i **list**, pa je tu javno nasleđivanje na mestu. Na drugu stranu, može da se napravi i sistem u kome će **list** da nasleđuje **dlist**. Recimo da jednostavno nasledim klasu i onda „ubijem" one funkcije za kretanje unazad po listi. Šta onda? To mi se čini još lakše jer put od **list** do **dlist** ipak zahteva neki rad. Na drugu stranu ograničavanje **dlist** da bi se dobio **list** je trivijalan posao. *Of course*, to ne bi smelo da bude javno nasleđivanje, privatno bi tu bolje čučalo. Ajd javi se.

Biip

Dejan trenutno nije kod kuće. Ostavite poruku

Tanja ovde. Ajde, javi mi sa da se dogovorimo da mi instaliraš *Matematiku*. Videla sam je na faksu i oduševila sam se. Još samo da nabavim džepni PC i završila sam fakultet. Zezam se. Ono što me je oduševilo kod programa je to što u matematiku uvodi eksperiment. Ako pogledaš sve „normalne" nauke, ljudi u njima eksperimentišu i onda na osnovu toga prave neke zaključke. Klasična matematika, naravno, izvrće stvari na sasvim drugu stranu – prvo smisliš teoremu i onda se trudiš da je dokažeš. A ono što je super u *Matematici* jeste to što ti omogući da relativno brzo napraviš niz eksperimenata da proveriš da li si u pravu, pa da tek onda upotrebiš klasičan pristup. Ono što me plaši jeste da će klinci skroz zaboraviti da računaju. Evo ja recimo zbog digitrona uopšte ne znam da izvadim koren iz broja iako studiram matematiku. Dok mi je ne instaliraš idem da pičim *Tetris*. Ćao!

Biip

Dejan Jelović

Dejan trenutno nije kod kuće. Ostavite poruku

Deki, Saša na telefonu. Treba mi savet. Kako da dokumentujem objektno-orijentisane sisteme? Naime, što se tiče pojedinačnih klasa, to je lako – napišem šta klasa predstavlja, šta radi, i stavim komentare kod svih javnih i *protected* članova. Relativno slično dokumentovanju klasičnih strukturiranih programa. Ono što ne mogu da utvrdim jeste kako da dokumentujem različite interakcije među objektima? Čini mi se da je dobra ideja dokumentovanje kada i kako određeni objekti sarađuju, ali to već zalazi u dinamičke dijagrame, što me mrzi da radim. Usput, čini mi se da definisanje statičkih dijagrama koji predstavljaju odnose među klasama može programer da radi i sam jer se to svodi na puko poznavanje tipova informacija koje objekti treba da razmenjuju, dok je za dinamičke dijagrame potreban i ekspert u domenu problema koji je upoznat sa svim interakcijama koje se dešavaju. Kako se ovo uklapa u tvoju šemu? Javi se.

Biip

Dejan trenutno nije kod kuće. Ostavite poruku

Dejo, Bane na telefonu. U nekim od prošlih „Računara" pisao si o Unix-style radnom okruženju u kome će korisnik raznih aplikacija imati mogućnost iste integracije kao i neki Unix programeri sa svojim alatkama. Elem, to ima i ima OOI, ili *Document Oriented Interfaces*. Recimo, imaš tekst procesor. E sad to nije tekst procesor kao što je *WordPerfect*, već podržava samo glupi unos teksta i različite fontove. Sada ti na njega kačiš različite alatke: provera pravopisa, *search & replace*, štampanje i tako dalje. I kada ti treba da recimo tražiš po regularnim izrazima, ti samo zameniš alatku za pretragu, a sve ostalo ostaje isto. Ili ti treba srpski *spelling checker* umesto engleskog? Samo zameniš odgovarajuću alatku. Kul, a? Najočija fora u svemu tome jeste što su sve ove „alatke" koje čine tvoju aplikaciju zapravo objekti. Ima li boljeg načina da se spoje dve komponente? Recimo, napravi se apstraktna klasa **tekst_procesor** i onda se recimo objekti za štampanje prave da rade sa njom. E sada, ako ti umesto nje ubaciš njenog naslednika **word_perfect**, sve šljaka OK. Razmišljam da pravim operativni sistem koji bi podržavao sve to kao mali akademski projekat, pa se javi ako te zanima. Ništa komplikovano, zapravo se objektni pristup svodi samo na kreiranje jasno definisanog API-ja za svaku aplikaciju. Posle možemo da izazovemo IBM na dvoboj čisto da se utvrdi šta je bolje – naš OS ili ono njihovo polovče od operativnog sistema. Budi pozdravljen.

Biip



Dejan trenutno nije kod kuće. Ostavite poruku

De si rode! Imaš li neki pametan način da se simuliraju izuzeci u MSC 7.0? **Assert** makro mi u principu radi posao, ali ima tri velike mane.
Prvo, kao što znaš, svaka funkcija treba da ima set preduslova i set postuslova koji trebaju da budu zadovoljeni da bi proglasili da sve šljaka OK. E sada, ako preduslovi nisu OK, to jest ako program po ulasku u funkciju već ima korumpirane podatke, **assert** će da prekine izvršenje u toj funkciji umesto da se vrati u onu funkciju koja ga je pozvala, jer je tu zapravo greška. Drugo, treba da postoji način da se proveravaju pre- i postuslovi za celu klasu, što bi se radilo samo za pozive „od spolja", dok bi interni pozivi ostali bez ovih provera. I najgore, svaki **assert** biva ugrađen u kod i nasleđen. Sada umesto klasičnog „preduslovi se nasleđivanjem mogu olabaviti a postuslovi se mogu pojačati" imamo samo pojačavanje i pre i postuslova. Bljak. Nekako nazirem da je rešenje u finijoj granulaciji objekta i ubacivanjem **assert** makroa u posebne virtuelne funkcije, ali bi to načinilo užasne rugobe od svake klase koju napraviš.
Ne znam šta je gore. *By the way*, implementirao sam onaj skupljač đubreta na C++-u, ali da bi bio efikasan on ne koristi klasične pointere, već je programer primoran da umesto njih koristi klase **charptr**, **voidptr**, **intptr** itd. Sada još samo da preradim ovih trideset i kusur megabajta koda koji sam napisao tokom godina da radi s njim i završio sam stvar. Vidimo se kroz jedno pola veka.

Biip

Dejan trenutno nije kod kuće. Ostavite poruku

Dejane, Marko ovde. Vezano za objektno-orijentisano programiranje i kineski, ne čini li ti se da su sve paradigme koje smo do sada koristili zapravo samo implementacije naših ideja iz prirode. Recimo, postoje gliste čija tela idealno oslikavaju iteraciju. Drveće – prirodni način da ostvari rekurziju. Praktično svaka paradigma koju smo ikada izmislili već je postojala u prirodi. Čini mi se da bismo trebali da pretražimo biljni i životinjski svet da bismo videli da li je preostala još neka moćna filozofija koju smo propustili. Pade mi napamet, ovo može da bude apsolutno najbolji način da se isforsira objektni fazon – pričaš, recimo, sa direktorkom računskog centra koja uporno odbija da vidi istinu o eleganciji objektnog pristupa, i ti joj jednostavno kažeš „Dobro, kako vidim da vam se OOP ne sviđa, jasno je da ćete, da biste ostali produktivni, morati da odete korak dalje u ameboidalno programiranje." Ili recimo programiranje zasnovano na strukturi buba-švabe. Kladim se da bi žena u tom trenutku prihvatila i da proda dušu đavolu samo da to izbegne. Hmmm, možda zato ne treba stavljati žene na odgovorna mesta?

Biip

Dejan trenutno nije kod kuće. Ostavite poruku

Sandra ovde. Čuj, onaj tvoj ortak Miljan širi glasine po Sezamu da ti zapravo pišeš o objektnom programiranju tako što prepisuješ poruke sa sekretarice. Još je gora glasina da si ti zapravo samo *front-man* za Milana sa Matematičkog fakulteta, koji u stvari piše sve tekstove. Navodno su oni puni muškog šovinizma, što tebi nije svojstveno, jer ti zapravo nikada nisi ni video ženu. Inače, pojavi se ponekad i noću na Sezamu. Nema ništa lepše od malih noćnih razgovora. Pusti te objekte i programiranje.

Biip

AKO VOLITE MIŠEVE

Posle tastature, koju smo detaljno obradili pre nekoliko meseci, na red dolazi neizbežni miš. Danas se koristi u svakom ozbiljnijem programu i nema razloga da se i tim uređajem ne pozabavimo.

Miljan Jovanović

I pored očiglednih prednosti moram priznati da sam oduševljen ovim uređajem. Možda je razlog to što sam komunikaciju sa računarom počeo preko tastature (DOS) ili želo što na stolu nemam dovoljno mesta da se razmahnem i poteram kursor po ekranu. Takođe bitan razlog za tako malu upotrebu je i to što ne volim razne grafičke korisničke aplikacije (kao *Windows* npr. – operativni sistem?) gde je upotreba miša gotovo obavezna. Ne bih hteo da iznosim posebne detalje protiv upotrebe grafike pa neka ostane na tome da ne vidim bitnu prednost (osim u izuzetnim slučajevima), a ozbiljniji program bi zahtevao bar 386/VGA sistem. Sa druge strane, upotreba miša u grafičkom režimu uglavnom nije „isplative" van sisteme koji radi u grafici (OS/2), što izlazi iz okvira ovog teksta. Zbog toga će ovde biti reči o upotrebi miša u tekst modu i o tome kako postići ono što se može videti u ozbiljnim programima.

KONCEPCIJA

Veznik za miša pruža programeru tridesetak funkcija čiji je delimičan spisak prikazan u tabeli 1. Sve funkcije se pozivaju preko interapta 33h, pri čemu AX registar treba da sadrži vrednost (broj) funkcije. Pre njihove upotrebe obavezno treba ispitati prisustvo veznika funkcijom 00h – ako je miš propisno instaliran, onda će u AX registru biti vraćena vrednost -1, u drugom slučaju će biti 0. (*Napomena: Tekst koordinate X i Y se moraju pomnožiti sa 8 ako se prenose kao parametar nekoj funkciji, i koordinate dobijene od veznika treba podeliti sa 8 da bi se dobile ispravne vrednosti za tekst režim.*)

Koncepcija treba da obezbedi prevazilaženje potencijalnih problema vezanih za same upravljače, ali tako da time ne ugrozi sam kvalitet programa. Najjednostavniji način upotrebe miša pruža već postojeća funkcija 03h. Ona obezbeđuje da u svakom trenutku možete saznati položaj kursora i stanje tastera, što može biti dovoljno u jednostavnim situacijama. Šta ovde predstavlja problem? Problem je ako želite „razvlačenje" kursora po ekranu (pomeranje miša uz pritisnut taster) ili ako vam za potvrdu neke ređe operacije treba dupli klik. To ove funkcija ne može uraditi, zato problem rešavamo na drugi način: od upravljače koristimo samo one najvažnije rutine za obradu informacija pristiglih od miša – komunikacija sa uređajem, a sami pišemo sve ostalo.

Svaki standardni upravljač za miša ima mogućnost da pri nekom događaju (pomeranje miša, pritisak ili otpuštanje tastera) pozove eksternu rutinu i da joj se saopšti koji je razlog za njeno pozivanje. (U literaturi ime ove rutine je *Event Handler* (EH), ali pošto joj nisam našao prikladan prevod odlučio sam se samo za engleski naziv.) U dokumentaciji je navedeno da se instalacija ove rutine obavlja na dva načina: preko funkcija 0Ch ili 18h (ax= fn int 33h). Razlika između ove dve funkcije je u nešto proširenijem setu događaje koji je aktiviraju u korist funkcije 18h (prati još i stanja **Shift, Alt i Ctrl** tastera). Međutim, kako smatram da stanje ovih tastera nije teško ispitati ako je potrebno, odlučio sam se za onu jednostavniju varijantu.

Postavljanje EH vrši se ovako: AX treba da sadrži broj funkcije (0Ch), CX sadrži masku događaja – uslov za pozivanje EH:

bit 0 – pomeranje miša
bit 1 – levi taster pritisnut
bit 2 – levi taster otpušten
bit 3 – desni taster pritisnut
bit 4 – desni taster otpušten
bit 5 – srednji taster pritisnut
bit 6 – srednji taster otpušten

Sve događaje obuhvatamo ako u CX registar postavimo 7Fh (ukidanje poziva EH CX= 0000h). Naravno da treba predati i adresu same EH rutine – to se vrši preko registara ES:DX (seg:off). Znači ako pre pozive ove funkcije u CX registar postavimo vrednost 00000110b, to će se EH aktivirati samo pri promeni stanja levog tastera. Kako nas zanima svaka promena, onda će CX sadržati 01111111b.

Mala komplikacija je što veznik za miša pri pozivu EH rutine postavlja neke vrednosti u registre koje moramo sačuvati, a između ostalog i DS registar pokazuje na segment sa *njegovim* podacima (*data segment*). Da bi mogli koristiti naše promenljive na samom početku rutine moramo prvo postaviti ispravnu vrednost u DS – razlog više za malo asemblera. Evo kako su raspoređeni parametri koje veznik prosleđuje našoj rutini:

AX – maska koja definiše tekući događaj
BX – stanje tastera **bit 0** = levi taster
bit 1 = desni taster
bit 2 = srednji taster
CX – X koordinata kurzora
DX – Y koordinata kurzora
SI – horizontalno rastojanje od poslednjeg pomeranje (<0 – levo >0 – desno)
DI – vertikalno rastojanje od poslednjeg pomeranja (<0 – gore >0 – dole)
DS – segment sa podacima veznika

Kao što možete primetiti, u ovim podacima se nalaze i koordinate tekućeg položaja kursora miša – zašto ne bismo njima popunili neke globalne promenljive i tako u svakom trenutku imali tačne koordinate? Nema argumenata protiv, pa će se koncepcija na tome i zasnivati. Zato definišemo jednu globalnu strukturu:

```
struct MOUSE_DATA {
   volatile int dugmed,// stanje dugmedi
   volatile int x,          // x koordinata
```

Event Handler

```
   ...
     static ...
     static ...
     int x, y;
     word event
     long clock_count

   asm {     mov word ptr event, ax
             mov ax, cx        // x= x/8
             shr ax, 1
             shr ax, 1;
             shr ax, 1
             mov word ptr x, ax
             mov ax, dx        // y= y/8
             shr ax, 1
             shr ax, 1
             shr ax, 1
             mov word ptr y, ax
             mov ax, seg _DATA                    // postavljanje DS= DATA
segment
             mov ds, ax

             mov ax, WORD ptr mouse_d.dugmed      // dugmed= (dugmed & 0xfff0) |
bx
             and ax, 0FFF0H
             or ax, bx
             mov WORD ptr mouse_d.dugmed, ax

             xor ax, ax                           // clock count= disktime(0
0)
             int TIME_INT
             mov WORD ptr clock_count, dx
             mov WORD ptr clock_count+2, cx   }

   if(event & 0x02) {                    // levi taster pritisnut
      tl= clock_count - tl;
      asm   xor WORD ptr mouse_d.dugmed, LEFT_PRESSED
      mouse_d.shape= 1
    #pragma warn -rng
```

```
        if(tl >= DOUBLE_CLICK_MIN && tl <= DOUBLE_CLICK_MAX) {
      #pragma warn +rng
          tl= 0
          asm or WORD ptr mouse_d.dugmed, LEFT_DOUBLE;
          mouse_d.shape= 3
        }
     }
     if(event & 0x04) {                         // levi taster pušten
        tl= clock_count
        asm   not word ptr mouse_d.dugmed, LEFT_DOUBLE
        mouse_d.shape= 0
     }

     if(event & 0x08) {                         // desni taster pritisnut
        tr= clock_count - tr
        asm   xor word ptr mouse_d.dugmed, RIGHT_PRESSED
        mouse_d.shape= 2
      #pragma warn -rng
        if(tr >= DOUBLE_CLICK_MIN && tr <= DOUBLE_CLICK_MAX) {
      #pragma warn +rng
          tr= 0
        asm   or WORD ptr mouse_d.dugmed, RIGHT_DOUBLE
        mouse_d.shape= 4
      }
   }

   if(event & 0x10) {                         // desni taster pušten
      tr= clock_count
      asm   not word ptr mouse_d.dugmed, RIGHT_DOUBLE
      mouse_d.shape= 0;
   }

   if(mouse_d.status)                         // obriši kurzor
      mouse_print(0)
   mouse_d.x= x;                              // promeni koordinate
   mouse_d.y= y
   if(mouse_d.status)                         // iscrtaj kurzor
      mouse_print(1)
}
```

Kontrola miša

Fn	Opis
00h	Provera da li je instaliran i resetovanje miša
02h	Skrivanje kurzora (isključivanje)
03h	Položaj kurzora i stanje tastera
04h	Podešavanje X i Y koordinate (nov položaj kurzora)
07h	Podešavanje granica za promenu X koordinate
08h	Kao i Fn 07h za Y koordinatu
0Bh	Horizontalno i vertikalno rastojanje od poslednjeg poziva
0Fh	Podešavanje brzine kurzora
13h	Podešavanje ubrzanja kurzora
14h	Zamena Event Handler (EH) rutine Nove <=> Postojeća
15h	Potrebna veličine memorije za čuvanje statusa
16h	Snimanje statusa
17h	Vraćanje snimljenog statusa

Tabela 1: Delimičan spisak funkcija za kontrolu miša (Int 33h)

```
   volatile int y,        // y koordinata
   volatile int status,   // vidljiv kurzor ili ne?
   volatile int postoji,  // instaliran? 0 -ne, 1 -da
   volatile int shape     // izgled kurzora
};
```

Potrebno je deklarisati samo jednu ovakvu strukturu EH će pri svakom pozivu podesiti njene članove. U primeru se ona naziva **mouse_d**, a definisani makro MOUSE je u stvari skraćenica za člen „postoji" i može se koristiti za ispitivanje postojanja miša. Sadržaj polja **status** govori o tome da li je kursor vidljiv ili ne, dok **shape** predstavlja izgled kursora. Dakle **Event Handler** treba da obradi sledeće: podešavanje X i Y koordinata, stanja tastera, dvostruki klik i štampanje kursora.

Nekoliko makroa definisanih u **mousedrv.h** treba koristiti za ispitivanje stanja tastera u programu. Ako želimo da saznamo da li je levi taster pritisnut:

```
if(mouse_d.dugmad & LEFT_BUTTON)
   printf(„L"),
```

Na sličan način treba koristiti **RIGHT BUTTON, LEFT DOUBLE** i **RIGHT DOUBLE,** dok se **LEFT_PRESSED** i **RIGHT PRESSED** koriste malo drugačije: vrednosti bitova na njihovim mestima se menjaju samo pritiskom na taster, tako da njihovim čitanjem možete odrediti da li je došlo do promene stanja tastera.

```
if(mouse_d.dugmad & LEFT_BUTTON) {
   word L= mouse_d.dugmad & LEFT_PRESSED,
   while(L == mouse_d.dugmad & LEFT_PRESSED)
      move(mouse_d.x, mouse_d.y);
}
```

Treba primetiti da se ovim rešenjem olakšava uzimanje tekućih koordinata i statusa tastera, jer se oni ažuriraju svakom promenom, pa su pristupačni bez ikakvih funkcija.

EH rutini je poveren još jedan važan zadatak – prikazivanje kursora na ekranu. Razlog za koncepciju ovakvog rešenja leži u tome što mnogi upravljači ne rade ispravno za tekst rezolucije preko 80x25. Primer vezan za ovo je dosta jednostavan i nije nikakav problem prebaciti ga iz C-a u asembler radi eventualno veće brzine štampanja.

Izvestan deo funkcije **mouse_event()** (slika 2) pisan je u asembleru zbog ispravnog preuzimanja parametara koji su prosleđeni kao sadržaj registara. Drugi deo nije bilo neophodno pisati u asembleru jer je Borlandov C prevodilac sasvim dobro obavio optimizaciju koda, tako da sam taj deo prepustio njemu. Promenljive se iz ove rutine mogu koristiti samo *posle* podešavanja DS registra, ali se za to lokalne promenljive mogu koristiti i pre toga (uz dužnu pažnju da sadržaji registara ostanu očuvani).

OBJEKT MOUSE

Sve važnije funkcije su zavijene u objekat **Mouse** koji pre upotrebe treba definisati a time i startovati konstruktor. Moramo poštovati ostale programe koji koriste miša pa je zato potrebno snimiti stanje upravljača pre nego što sami učinimo bilo kakve promene. Ovo snimanje se svodi na kopiranje internih parametara upravljača u neki deo memorije – posao za konstruktor.

Veličina potrebne memorije nije definisane pa se one dobija pomoću funkcije 15h, a snimanje funkcijom 16h. Pored ovoga konstruktoru je dodeljeno podešavanje EH rutine i još nekih podataka.

Sledi kratak opis funkcija.

Funkcija **Mouse install()** se koristi samo na početku za proveru da li je miš instaliran, jer provere nije potrebne tokom izvršavanja programa. Međutim, ta funkcija se koristi i za resetovanje upravljača, tako da je ostala u *public* delu objekta.

Funkcije **show()** i **hide()** vrše samo dotervanje promenljive **status** u strukturi **MOUSE_DATA**. Tu promenljivu koristi funkcija koja vrši štampanje/brisanje kursora na ekranu. Ako je podešena na TRUE, kursor će biti prikazan, u suprotnom obrisan.

Mouse set(x, y) će podesiti položaj kursora na vrednost koju zadate. Nije preporučljivo koristiti je (osim u izuzetnim slučajevima) jer se time korisnik može zbuniti. Na primer, za pozicioniranje kursora u gornji desni ugao ekrana možete pozvati **set(Xmax-1, 0)**, baš kao u funkciji **home()** – zgodno je ovo uraditi na početku programa da bi se izbeglo pojavljivanje kursora na sredini ekrana.

Tri funkcije **Xmotion()**, **Ymotion()** i **mrd()** treba koristiti kada želimo da saznamo da li je došlo do pomeranja kursora (od prošlog poziva) ili ne. U principu, ovde je najznačajnija funkcija **mrd()** koja vraća TRUE ako je bilo nekakve promene (položaj kursora ili stanje tastera) od prošlog poziva. Ako imate nameru da u svoj program ugradite neki *screen saver*, onda ovom funkcijom možete saznati da li je korisnik aktivan ili ne.

Problemi vezani za primer opisan u tekstu su se uglavnom nalazili u samom upravljaču za miša – u **mouse.sys** (ili omiljenioj TSR verziji **mouse.com**). Pokušaj da rešim problem uzimanjem drugog upravljača nije urodio plodom, već sam imao prilike da se sretnem i sa drugim „bubicama". Naime, originalni upravljač (koji sam dobio uz miša) ne radi ispravno sa tekstualnim rezolucijama većim od 80 kolona i 43 reda, koje volim da koristim. Jednostavno ne ume da izračuna tačan položaj kursora na ekranu, već i dalje smatra da je širina 80 znakova (a ne 132) itd. Takođe nisam verovao da ni obično pozicioniranje kursora u gornji levi ćošak ne radi, pa sam funkciju **home()** prekrstio u gornji desni ćošak. Što se tiče ovoga, u razgovoru sa jednim prijateljem sa Sezama saznao sam da njegov upravljač inicijalizaciju radi nekoliko sekundi? Sve ovo govori da problema ima prevše i da se programer mora osloniti samo na neophodne stvari.

Kompletan listing ovih rutina se nalaze na Sezamu, pa svi koji žele mogu ga preuzeti iz R:\RSOFT direktorijuma.

Sa autorom se može kontaktirati na *Sezamu* preko pseudonima **mjova**

Nastavak sa strane 59

preko neke druge (u najboljem slučaju), ili preko nečeg drugog (recimo steka).

Maksimalan niz za **far** pokazivače je veličina 65531 a ne 65536! Četiri bajta koji nedostaju koriste se kao zaglavlje **heap**-a, a pomenuta okrugla vrednost je, naravno, posledica pomenute ograničenosti na jedan segment.

Neobraćanje pažnje na vreme postojanja varijable je još jedna standardna greška. Recimo, pokušaj povratka lokalne varijable iz funkcije kao njenog rezultata može raditi neki put, jer je ona fizički možda još prisutna u memoriji, među ne kao deo steka gde je originalno bila smeštena. Može se međutim desiti da se po izlasku iz te funkcije, kada se dotična varijabla skida sa steka, na stek odmah stavlja neki drugi podatak, koji će se prepisati preko ovog našeg, i tek tada će se greška manifestovati.

Rad sa stringovima koji nisu završeni nultim karakterom je takođe česta greška, a može se javiti i u kombinaciji sa indeksiranjem van granica (recimo, nulti karakter postavimo u **niz[50]**, a niz smo definisali kao **niz[50]**). Sve funkcije iz **string.h** se oslanjaju na postojanje ovog specijalnog karaktera u radu, a njegov nedostatak će za većinu njih značiti da rade dalje sve dok ne naiđu na njega. Zato je, između ostalog, dodata i „n" familija ovih funkcija (**strncpy, strnicmp,** i slične), koje će operisati u zadatom prostoru sve dok ne naiđu na nulti karakter ili zadatu granicu prostora.

Veličina steka nikad ne bi trebalo da bude postavljena na više od 0xFFE0 pomoću **_stklen**, jer kompajler automatski poravnava stek na paragraf (16 bajtova). Ova veličina će zato postati po kompajliranju 0xFFF0, a da smo postavili **_stklen** na više od 0xFFEF, kompajler bi je pramotao na 0!

Jedna od najmisterioznijih grešaka je svakako „Null Pointer Assignment", koja se pojavljuje redovno po izlasku iz programa. Generišu je programi koji kvare dno **data** segmenta, a moguća je samo u **small** i **medium** modelima. Borland, naime, na dnu tog segmenta postavlja četiri nule praćene porukom. U **small** i **medium** modelima, NULL pokazivač je **near** tipa, pa pokazuje na DS:0000h, što je zapravo prvi bajt **data** segmenta. Na izlasku iz programa proverava se postojanje ta četiri bajta i **copyright** poruka, i u slučaju bilo koje modifikacije izbacuje se pomenuta greška. Ovo je, dakle, posledica već objašnjenog korišćenja neinicijalizovanih pokazivača (što je i NULL pokazivač). U **far** modelima, da ponovimo, NULL je adresa 0000:0000h, tako da će **data** segment ostati netaknut, ali će nestradati zona pokazivača na sistemske prekide. U **tiny** modelu sva četiri segmentna registra pokazuju na isti segment, tako da će stradati početak koda, a pitanje je možemo li uopšte osetiti i kada. Da naglasimo, greška se može pojaviti i kada je pokazivač inicijalizovan, ali na pogrešnu adresu (tačnije, ako slučajno pokazuje baš negde u pomenuti zaštićeni prostor u **data** segmentu).

I tako dalje...

EPILOG

Ovo je očigledno bila mala, malecka škola o memorijskim modelima i problemima sa memorijom. Moglo bi se tu još dosta pričati, ali autor smatra da je ipak najbolja škola praksa, i samo praksa. Recimo, kada vam program tvrdoglavo padne sedmi put uzastopce, a sve na prvi pogled izgleda OK, pa onda provedete celu noć pred monitorom tražeći grešku, nijedna knjiga neće moći da nadomesti znanje i iskustvo koje ste stekli. Rad, rad i samo rad, a ponekad i košarka.

JEDAN PRAVI TSR

Mada smo već više puta govorili o pisanju rezidentnih programa, ovaj put ćemo se potruditi da ne pričamo baš iste stvari. Naime, pokušaćemo da napravimo neku vrstu šablona za pisanje TSR programa, koji će biti sasvim dovoljan za skoro sve potrebe. Rezidentni programi rađeni njime će moći da se probude bilo pritiskom na taster, bilo posle nekog vremenskog perioda, da se „dižu" u visoku memoriju, da se deaktiviraju i brišu iz memorije i tako dalje. Istovremeno, videćemo kako se memorijski modeli i segmenti, kojima posvećujemo poseban članak u ovom broju, mogu iskoristiti za ovu svrhu.

Rezidentni programi su jednostavno posledica prirode operativnog sistema DOS. Naime, takav kakav je, DOS ne dozvoljava istovremeni rad više programa, što je korisnicima često veoma potrebno. Zato se neko dosetio da iskoristi postojanje prekida i nekih zgodnih funkcija sistema, i napravi program čiji će deo (ili ceo program) ostati aktivan u memoriji paralelno sa drugim programima.

Uloga rezidentnih programa može biti veoma trivijalna: klik pri pritisku na taster, sprečavanje resetovanja sistema klasičnom kombinacijom tastera, sat u uglu ekrana. Međutim, postoje veoma komplikovani programi ove vrste koji koriste prozore i menije, ispisuju čitave gomile podataka, pale se i gase na razne načine pa čak i sami sebe brišu iz memorije kada više nisu potrebni. Sa korisničke strane, jedina sličnost među njima je činjenica da su rezidentni i da se većina poziva ili pritiskom na neku čudnu kombinaciju tastera, ili na osnovu nekog brojača. Mi ćemo pokušati da napravimo jedan šablon („stručno" rečeno, *template*) koji će omogućiti pisanje bilo onih trivijalnih, bilo najkomplikovanijih rezidentnih programa, sa svim njihovim dobrim i lošim stranama.

ŽIVOT I PRIKLJUČENIJA

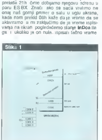
Mogli bismo ovde početi sa nekim programčićem koji će proizvoditi manje ili više potreban klik pri pritisku na taster, ali smatramo da je o rezidentnim programima toliko bilo reči, da bi to bilo suvišno. Činjenica je da bi takav program zapravo samo preuzimao jedan prekid i ostao aktivan u memoriji. Kada bi korisnik pritisnuo taster, što bi dotični prekid javio programu, ovaj bi generisao zvuk. I to je sve.

Međutim, ovde smo već zapazili sledeće dve važne stvari: rezidentan program preuzima neki prekid koji će mu javiti kada da „oživi" i ostaje aktivan u memoriji. O ovoj drugoj osobini ćemo na kraju ovog odeljka, a sada ćemo se pozabaviti preuzimanjem prekida. Postoji bar pet prekida koji mogu biti korisni nekom rezidentnom programu (u daljem tekstu TSR, od „Terminate and Stay Resident"), a neki se mogu svrstati u hardverske prekide PC računara. Prvi je svakako prekid 08h (*System timer*) koji se aktivira 18.2 puta u sekundi. Vezujući se na njega, bez problema možemo realizovati ranije pomenuti sat u uglu ekrana: kad god sistem aktivira ovaj prekid, aktiviraće se i naš TSR koji će preračunati vreme i, ako je potrebno, ispisati ga u uglu ekrana. Ispisati kako? Naime, klasičan pokušaj ispisivanja korišćenjem standardnih C funkcija će vrlo verovatno oboriti sistem, na prvi pogled bez razloga. Dublji pogled na ove funkcije pokazaće da one koriste pozive standardnim DOS funkcijama prekida 21h. E, tu se vraćamo na onu opasku o prirodi operativnog sistema DOS.

Sve DOS funkcije su projektovane tako da se mogu pozivati samo jedna po jedna, na osnovu ideje da će u datom trenutku biti aktivan samo jedan program i da on, iz očiglednih razloga, u jednom trenutku može pozvati samo jednu funkciju. One pri radu koriste bar jedan od moguća tri interna steka: IO stek za funkcije 01h do 0Ch, *Disk* stek za gotovo sve preostale funkcije i *Auxiliary* stek za kritične greške. Pri prvom pozivu nekoj DOS funkciji ona prebacuje „pogled" sistema sa programskog na odgovarajući interni stek. Ako sada, tokom rada ove funkcije, pokušamo da pozovemo neku drugu koja koristi isti interni stek, ona će ga ponovo inicijalizovati i time potpuno zbrljati informacije na steku koje je zapisala ona prva funkcija. Ovde se, dakle, nameću dva rešenja: ili uopšte ne koristiti DOS funkcije, ili registrovati kada je neka od njih aktivna, pa sačekati dok ne završi rad. Bez funkcija prekida 21h teško da se može uraditi bilo šta ozbiljno, tako da ćemo morati da se pozabavimo drugom mogućnošću.

Na našu sreću, DOS nudi rešenje, mada ne baš sasvim dokumentovano. Sistem poseduje jedan poseban fleg, veličine 1 bajt, čija je uloga da broji koliko je DOS funkcija trenutno aktivno. U većini slučajeva će njegova vrednost biti ili 0 ili 1, što je sasvim dovoljno da znamo kada možemo sa sigurnošću zvati bilo koju funkciju prekida 21h. Adresu ovog flega, znanog kao InDos, nalazimo funkcijom 34h prekida 21h, čime dobijamo njegovu adresu u paru ES:BX. Znači, ako se sada vratimo na onaj naš gornji primer o satu u uglu ekrana, kada nam prekid 08h kaže da je vreme da se aktiviramo, a mi zaključimo da je vreme ispisivanja na ekran, pogledaćemo stanje **InDos** flega, i ukoliko je on nula, ispisati tačno vreme.

Slika 1



Struktura programa u *large* modelu

Ideja je dobra, ali nije sve baš tako lako.

DOS ima još jednu (od više njih) nezgodnu osobinu: kada nastupi kritična greška, on očisti **InDos** fleg i ulazi u neko nestabilno stanje, kada se samo veoma mali broj funkcija može sa sigurnošću pozvati. Znači, može se desiti da **InDos** bude nula, ali da opet ne smemo da koristimo DOS funkcije. Zato nalazimo još jedan fleg, znan kao **Critical Error** fleg, koji je takođe jednobajtna veličina. Ukoliko je nastupila kritična greška, **InDos** će biti nula, a ovaj fleg jedan. Položaj ovog flega je odmah posle **InDos**, što znači da nam ona funkcija 34h daje oba. Ovde sledi i jedna napomena: pošto one takođe spadaju u DOS funkcije, adrese flegova moramo pronaći pre nego što aktiviramo naš TSR, a ne iz njega. Uzgred, postoji još jedan način nalaženja ovog flega, koji ćemo mi koristiti, a to je funkcija AX=5D06h, kao što je to demonstrirano u tekstu programa (modul INIT.C).

Sada bi se reklo da smo sve sredili, ali nismo. Šta mislite koja funkcija „radi" kada ste na komandnoj liniji? Naravno, funkcija 0Ah prekida 21h koja čita tastaturu. Eto opet: **InDos** će, dok god se nalazimo na komandnoj liniji, sadržati jedan i naš sat neće biti ispisan. Ali DOS opet ima keca u rukavu: dok se čeka na komandnoj liniji, stalno se poziva prekid 28h, poznatiji kao „Dos Idle" prekid („Dosadno mi je" prekid). Ako je u tom trenutku **InDos** jedan, **Critical Error** fleg nula, možemo sa sigurnošću koristiti DOS funkcije (DOS ništa ne radi, jel'), ali samo one od 0Dh naviše. Zašto samo njih? Zato što čekanje na komandnoj liniji da ponovimo, poziva funkciju 0Ah, koja koristi, baš kao i sve one ispod 0Dh, isti interni IO stek. A iste te funkcije koristi i većina procedura za ispisivanje na ekran. Eto li ga sad, džaba smo se mučili?

A, ne. Cela ova priča, logično, i nije bila namenjena satu u uglu ekrana (ispis bi zapravo radili „ubadanjem" potrebnog teksta u taj prokleti ugao), nego onim složenijim slučajevima kada ne možemo izbeći pozivanje DOS funkcije. Pomenute kritične funkcije prekida 21h (one do 0Dh) se brinu za izlaz na ekran, printer i očitavanje tastature, i umesto njih se mogu koristiti i neki BIOS prekidi i sopstvene rutine, ali one sa višim rednim brojem su često nezamenljive.

Da se sada vratimo na ranije pomenuti broj od „bar pet" prekida koje TSR može da koristi. Dva smo već pomenuli, a evo ga i treći. Naime, recimo da naš ne možemo deaktivirati kada poželimo, recimo pritiskom na neku kombinaciju tastera. E, ako obratimo pažnju na prekid 09h, koji se aktivira pri svakom događaju na tastaturi, imaćemo rešenje. Naš sat će preuzeti ovaj prekid i, kad god se aktivira, proveriti šta se dogodilo na tastaturi – ako je neka naša kombinacija tastera pritisnuta, daćemo signal satu da prestane da se ispisuje. Sad ka-

Prekidi.c

```
#include <stdlib.h>
#include <dos.h>
#include <string.h>
#include <dos.h>
#include "prekidi.h"

unsigned int tick_count        = 0
unsigned int tick_max          = 0
unsigned int hotshift          = 0
unsigned int hotmask           = 0
unsigned int im_inside         = 0
unsigned int want_inside       = 0
unsigned int disk_active       = 0

unsigned int new_stack[STACKSIZE] = {0};
unsigned int new_ss
unsigned int new_sp

char far *indos
char far *critical
unsigned int old_psp;
unsigned int olddtaseg  olddtaofs
unsigned int dtaseg  dtaofs

// string za identifikaciju svih TSR rodjenih ovim sablonom
// za svaki pojedini TSR se dodaje dodatna identifikacija MY_ID
// definisana u INIT H
char far* TSRNAME= TSR template v1.0 (C) Copyright 1993  Petrovic Nikola

void interrupt (far *old 8) (void)
void interrupt (far *old 9) (void)
void interrupt (far *old 13) (void)
void interrupt (far *old 23) (void)
void interrupt (far *old 24) (void)
void interrupt (far *old 28) (void)
void interrupt (far *old_2F) (void)


//
// Ulaz u TSR rutine sa parametrom inter i nasim stekom
//
void enter_tsr (unsigned int inter)
{
        struct REGPACK reg

        reg r_ax = 0x5100              // uzmi trenutni PSP
        intr(0x21  &reg)
        old_psp = reg r_bx,            // sacuvaj ga

        reg r_ax = 0x5000,             // postavi nas PSP
        reg r_bx = _psp
        intr(0x21, &reg)

        reg r_ax = 0x2f00              // nadji trenutni DTA
        intr(0x21  &reg)
        olddtaseg = reg r_es           // sacuvaj ga
        olddtaofs = reg r_bx

        reg r_ax = 0x1a00              // postavi nas DTA
        reg r_ds = dtaseg
        reg r_dx = dtaofs
        intr(0x21  &reg)

        old_23 = getvect (0x23)        // postavi nas CTRL break handler
        setvect (0x23, new_23)
        old_24 = getvect (0x24)        // postavi nas critical error handler
        setvect (0x24  new_24)

        tsr (inter)                    // !!! POZIV TSR RUTINAMA !!!

        setvect (0x24  old_24)         // vrati stari critical error handler
        setvect (0x23  old 23)         // vrati stari CTRL Break handler

        reg r ax = 0x1a00              // vrati stari DTA
        reg r_ds = olddtaseg
        reg r dx = olddtaofs
        intr(0x21  &reg)

        reg r_ax = 0x5000              // vrati stari PSP
        reg r_bx = old_psp
        intr(0x21  &reg)
}


//
// Novi tajmer prekid 08h
//
void far interrupt new 8 ()
{
        old_8 ()                            // pozovi prethodne originalni

        if ( (tick_max==0) &&  want inside) // sadrzaj nije definisan stog
                return                      // zbog tajmera i prethodno nije
                                            // pozvan nijedan  dogadjaj
        if ((++tick_count >= tick_max) || want_inside)
        {
                want_inside = 1
                tick_count = 0
                if (!im_inside  && (*indos==0) && !disk_active &&
(*critical==0))
                {
                        want_inside = 0     // vise ne zelim da se aktiviram
                        im_inside = 1       // jer upravo ulazim sa radom
                        new_ss = _SS        // tj( sopstveni stek
                        new_sp = _SP
                        _SS = FP SEG (new_stack)
                        _SP = FP OFF (&(new_stack[STACKSIZE]))

                        enter_tsr (8)       // krenu u rad

                        _SS = new ss        // stari stek
                        _SP = new sp
                        im_inside = 0       // vise nisam aktivan
                }
        }
}


//
// Novi keyboard prekid 09h
//
void far interrupt new 9 ()
{
        asm (sti)
        old_9 ()

        if (hotshift == 0 && hotmask == 0)      // mozda se trebe koristiti
                return                          // tastere za ulaz?

        if ((bioskey(2) & 0x0F)  != hotshift)   // kombinacija shift tastera
                return                          // nije odgovarajuca?

        if (hotmask == 0 || (bioskey(1) >> 8) == hotmask)
        {                                       // scan kod je odgovarajuci
                want_inside = 1                 // hocu da se aktiviram
                if (hotmask != 0) bioskey (0x10),
        }
}


//
// BIOS disk prekid 13h
//
#pragma argsused
void far interrupt new 13 (bp di si ds es,dx cx bx,ax ip cs,flags)
{
        disk_active=1           // kriticna disk operacija u toku!!
        old_13(),
        disk_active=0           // vise nije    u toku
}


//
// Novi DOS idle prekid 28h
//
void far interrupt new 28 (void)
{
        if (want inside == 1)
        {
                if (!im_inside && (*indos==1) && !disk active &&
(*critical==0))
                {
                        want_inside = 0,
                        im_inside = 1,
                        new ss = _SS
                        new sp = _SP,
                        _SS = FP SEG (new_stack),
                        _SP = FP OFF (&(new_stack[STACKSIZE]))

                        enter tsr (0x28)

                        _SS = new_ss
                        _SP = new_sp
                        im inside = 0
                }
        }
        else
                chain intr (old 28)
}
//
// Multiplex prekid 2Fh
//
#pragma argsused
void far interrupt new_2F (bp,di,si,ds es,dx cx bx ax ip cs flags)
{
        unsigned char ah,al

        ah=ax>>8        // identifikacija
        al=ax&0xff      // podfunkcija

        if ( (ah == MY_ID) && ( strcmp((char far *)(MK_FP(ds,si)),TSRNAME)==0))
        {
                if (al>=0x80)                   // komuniciraj
                {
                        ax=comunicate(si cx di)  // i vrati AX kao rezultat
                        return                  // AX=FFFFh za OK
                }

                switch (al)
                {
                        case DETECT CMD         // detekcija prisutnosti
                                si = 0xFFFF     // (skace to dex
                                bx = _psp       // moj PSP
                                break,
                        case ULGAS_CMD
                                if (  (FP_SEG(getvect (0x28)) ==
FP_SEG(new 28))
                                     &&(FP SEG(getvect (0x08)) == FP_SEG(new_8)
{
                                     &&(FP_SEG(getvect (0x09)) == FP SEG(new_9))
{
                                     &&(FP_SEG(getvect (0x13)) ==
FP_SEG(new 13))
                                     &&(FP_SEG(getvect (0x2F)) ==
FP_SEG(new 2F))
                                   )
                                {
```

```
                        ax = 0xFFFF
                        bx = _psp;
                        setvect (0x08, old_8)
                        setvect (0x09  old_9)
                        setvect (0x13  old_13)
                        setvect (0x2F  old_2F)
                        setvect (0x28  old_28)
                    }
                    else
                        si=0
                    break;
                default
                    _chain_intr (old 2F)
            }
            return
        }
    chain intr (old_2F)
}

//
// Novi ctrl Break prekid 23h
//
void far interrupt new_23 (void)
{
        // nemoj ništa da radiš za CTRL Break
}

//
// Novi critical error prekid 24h
//
#pragma argsused
void far interrupt new_24 (bp di si ds es dx cx bx ax,ip cs flags)
{
  ax = 3,        // odustani od toga što je izazvalo kritičnu grešku
}
```

Svi preuzeti prekidi

Zaglavlje sa potrebnim definicijama ◆

Prekidi.h

```
#define STACKSIZE 256     stack ...

#define MY_ID          0x6A
#define DETECT_...     0x00
#define UNLOAD ...     0x10

extern char *TSRNAME

extern void interrupt (far *old 8 ...
extern void interrupt (far *old 9 ...
extern void interrupt (far *old 13 ...
extern void interrupt (far *old 23) ...
extern void interrupt (far *old 24 ...
extern void interrupt (far *old 28 ...
extern void interrupt (far *old 2F ...

void interrupt far new 8 (
void interrupt far new_9 (
void interrupt far new 13 (int ...
void interrupt far new 23
void interrupt far new 24 (int ...
void interrupt far new_28 (
void interrupt far new 2F (int ...

void tsr  unsigned int
int communicate(char int int
void enter_tsr (unsigned int ...

extern unsigned int tick_count
extern unsigned int tick max
extern unsigned int hotshift
extern unsigned int hotmake

extern unsigned int new_stack[STACKSIZE]
extern unsigned int new ss
extern unsigned int new sp

extern unsigned int in_indos
extern unsigned int want insleep
extern unsigned int disk_int ...
extern unsigned int tsr_psp
extern unsigned int stackseg  oldstack
extern unsigned int stkseg  stkofs

extern char far *indos
extern char far *critical,
```

da smo konačno sve sredili, na vaše iznenađenje uopšte nećemo napraviti taj sat. Umesto njega, napravićemo šablon za pravljenje bilo kakvog rezidentnog programa što smo već nagovestili.

Spomenimo još i prekid 13h pod nazivom „BIOS diskette services" za koji je očigledno čemu služi. Pošto je reč o vremenski kritičnim operacijama sa diskelnim jedinicama ne bismo želeli da naš TSR upadne usred nekog upisivanja i prekine rad na pola puta ili još gore zbrlja podatke funkciji koja je u toku i uništi nešto na disku. Znači registrovaćemo poziv ovom prekidu i zabraniti rad našeg TSR dok se ne završi.

Konačno tu su i prekidi koji se generišu kada nastupi kritična greška ili korisnik pritisne kombinaciju **CTRL-Break**, ali o njima kasnije. Sada smo uglavnom obradili sve što je potrebno da jedan TSR proradi sem na početku pomenutog ostanka u memoriji. To se radi C funkcijom **KEEP**, koja opet zove jednu 21h funkciju. Parametri su status koji će program javiti pri povratku u DOS i veličina rezidentnog dela u paragrafima. Očigledno bilo bi interesantno da što manji deo programa ostane u memoriji a za to ćemo se poslužiti nekim sitnicama iznesenim u prethodnom broju „Računara" kada smo govorili o memorijskim modelima i segmentima. Ali, pre toga

JOŠ NEKE VAŽNE STVARI

Mada smo uspešno prošli kroz minsko polje DOS funkcija nismo baš o svemu vodili računa. Jedan TSR program nudi malo više uzbuđenje od izbegavanja funkcija 0Dh naniže i žongiranje sa flegovima.

Mnogi TSR programi umeju da čitaju ili pišu podatke na disk. Tu bi recimo spadale sve one rezidentne biblioteke, help sistemi rečnici i tako dalje. Mada zapisivanje podataka u datoteku izgleda kao trivijalna funkcija, a upravo smo se pobrinuli da ne uznemirimo DOS upadanjem u neki kritični proces za njeno realizovanje će nam biti potrebno još malo truda. Naime svaki aktivan proces pa i naš TSR podatke o otvorenim datotekama čuva u okviru zaglavlja memorijskog bloka koji mu pripada odnosno već pomenutom PSP (**Program Segment Prefix**). Ova struktura veličine 256 bajtova prethodi svakom bloku memorije o čemu smo već pisali (Računari 77). Dakle da bi naš TSR imao prave informacije o otvorenim datotekama i da ne bi otvaranjem ili zatvaranjem brljao po podacima nekog drugog aktivnog procesa pre ulaza u TSR moramo sistem obavestiti o postojanju našeg PSP a. To ćemo učiniti funkcijama 50h i 51h prekida 21h gde prva registruje novi PSP a druga saznaje koji je trenutno „aktivan". Verovatno je nepotrebno reći da po završetku reda naš TSR mora vratiti staro stanje stvari.

PSP ima još jednu interesantnu osobinu. Kada pozivamo funkciju 48h omiljenog nam prekida da alociramo blok memorije sistem će i za njega formirati PSP dodelivši mu novu oznaku o vlasniku (što ima svaki PSP) na osnovu procesa iz kojeg je funkcija pozvana. Ukoliko nismo obavestili DOS o promeni PSP pri ulasku u TSR novi blok će biti označen kao da pripada nekom aktivnom procesu čiji je PSP bio trenutno „važeći". Što se nas tiče blok će biti oformljen i moći će da se koristi ali samo dok postoji pomenuti proces kojem je blok pripao. U trenutku kada taj proces prestane da postoji DOS će potražiti sve blokove koji njemu pripadaju (sve one koji imaju istu oznaku o vlasniku u PSP) i obrisati ih iz memorije. Time bi dakle i naš blok zvanično nestao i eto problema. Usput da razjasnimo kad kažemo da se blok briše iz memorije ne mislimo na stvarno fizičko brisanje! Blok memorije se zapravo dealocira, odnosno promeni se podatak o njegovom vlasniku. Ukoliko blok ne pripada nikome dobija oznaku 0000h za vlasnika što je oznaka za sve blokove koji su slobodni za alociranje. Kada neka MEM komanda pokušava da sazna koliko imate slobodne konvencionalne memorije ona zapravo pronađe sve blokove čiji je vlasnik 0000h i obavesti vas o njihovoj ukupnoj dužini.

Kad smo već kod komunikacije sa datotekama pomenimo i funkcije 11h i 12h za pretraživanje direktorijuma (**FindFirst** i **FindNext** respektivno), koje koriste odgovarajuće funkcije C-a. Podaci o poslednjem nađenom fajlu, mestu na kojem smo zastali u pretrazi atributima kojima određujemo opseg pretrage i ostalo nalaze se zapisani u strukturi pod imenom DTA (*Disk Transfer Area*). Ako naš TSR želi da pretražuje direktorijume a da pri tom ne poremetimo neki drugi proces koji je možda upravo to isto radio moramo na ulasku u naš rezidentni program sačuvati podatke o trenutnom DTA obavestiti DOS da treba da koristi naš a po završetku vratiti onaj stari.

Stek programa je takođe važna stvar mada mu se malo pažnje posvećuje. Ukoliko naš program koristi i najmanji deo steka pri radu pametno je oformiti sopstveni stek dovoljne veličine. Ovo činimo ne samo zato što stek procesa koji smo prekinuli može biti premali za naše potrebe nego i zato što neki drugi TSR koji nas prekine može tražiti suviše steka za sebe a tvorac se nije setio da napravi svoj stek.

Konačno da rezimiramo šta treba uraditi: preuzeti prekide 08h 09h 28h 13h registrovati stanje **InDos** i **Critical Error** flegova oformiti sopstveni stek obavestiti DOS o promeni DTA i PSP a po završetku sve vratiti na staro. Ovako nabrojano i ne izgleda komplikovano?

ŠABLON

Došli smo do kritičnog dela ove rasprave. Sve što smo do sada izneli sada treba uobličiti u neki program i pokazati da radi. Krenimo dakle, od početka.

Svaki TSR ima deo zadužen za aktiviranje rezidentnog dela postavljanje potrebnih parametara očitavanje komandne linije ispisivanje pomoćnog teksta i tako to. Bila bi prava šteta (i nepotreban trošak) da sve to ostavimo da „visi" u memoriji kada program postane rezidentan. Da ovo radimo u asembleru bilo bi jednostavno program bi tako rasporedili da

Tsr.c

```
#include "prekid.h"

// argumenti prijo ... ju u kojima CX i DX izlaz AX

#pragma argsused
int communicate (unsigned int ax, int cx, int dx)
{
        return 0x0000;      // za sada uvek vracamo OK
}

// Telo vaseg TSR programa

void fa_tsr (unsigned int inter)
{
        ovde ide telo rezidentnog programa
        parametar inter kaze prekid koji je aktivirao TSR
}
```

Telo rezidentnog programa

init.c

```
#pragma option -zCTSR         // menjamo code segment klasu na TSR
                              // tako da dodje posle steka

#pragma option -zFTSR         // menjamo far data klasu na TSR
#pragma option -zETSR_DATA    // menjamo i segment ime

#include <stdio.h>
#include <dos.h>
#include "prekid.h"

// make kodovi za kombinacije tastera kojima se TSR proziva
// Kombinacije se podesavaju preko ALT+CTRL+KEY_X
enum key_make_codes
{
        NONE    = 0x0000
        ALT     = 0x0008,  CTRL    = 0x0400,  LSHIFT = 0x0200,  RSHIFT = 0x0100,
        KEY_A   = 0x001E,  KEY_B   = 0x0030,  KEY_C  = 0x002E,  KEY_D  = 0x0020,
        KEY_E   = 0x0012,  KEY_F   = 0x0021,  KEY_G  = 0x0022,  KEY_H  = 0x0023,
        KEY_I   = 0x0017,  KEY_J   = 0x0024,  KEY_K  = 0x0025,  KEY_L  = 0x0026,
        KEY_M   = 0x0032,  KEY_N   = 0x0031,  KEY_O  = 0x0018,  KEY_P  = 0x0019,
        KEY_Q   = 0x0010,  KEY_R   = 0x0013,  KEY_S  = 0x001F,  KEY_T  = 0x0014,
        KEY_U   = 0x0016,  KEY_V   = 0x002F,  KEY_W  = 0x0011,  KEY_X  = 0x002D,
        KEY_Y   = 0x0015,  KEY_Z   = 0x002C,  KEY_0  = 0x0078,  KEY_1  = 0x0078,
        KEY_2   = 0x007A,  KEY_3   = 0x007B,  KEY_4  = 0x007C,  KEY_5  = 0x007D,
        KEY_6   = 0x007E,  KEY_7   = 0x007F,  KEY_8  = 0x0080,  KEY_9  = 0x0081,
        K1      = 0x003B,  K2      = 0x003C,  F3     = 0x003D,  K4     = 0x003E,
        F5      = 0x003F,  K6      = 0x0040,  F7     = 0x0041,  K8     = 0x0042,
        F9      = 0x0043,  K10     = 0x0044,  SPACE  = 0x0039,  ENTER  = 0x001C,
        SYS_REQ = 0x0054,  TAB     = 0x000F,  ESC    = 0x0001
}

#define SUCCESS 0
#define CANT_UNLOAD 1
#define MEM_FROZEN  2
#define ENV_FROZEN  3

// poruke koje se ispisuju pri inicijalizaciji ili deaktiviranju odlaze
// ovde jer ce ovaj deo biti eliminisan pri aktiviranju TSR
char *message[]=
{
        "TSR unloaded",
        "TSR can't be unloaded",
        "TSR memory can't be deallocated",
        "TSR environment can't be deallocated"
}

// Kombinacija ALT klaster, CTRL klaster i SHIFT klaster se racunaju ovde
// drugacija nego ostale ... potrebna ispravka pri racunanju
int ALT_K   = 0x68   F1
int CTRL_F  = 0x5E   F1
int SHIFT_K = 0x54   K1

//
// Proveri da li je TSR vec u memoriji
//
int already_there (void)
{
        struct REGPACK reg;
        reg.r_ax = (MY_ID<<8)|DETECT_CMD
        reg.r_es = FP_SEG(TSRNAME)
        reg.r_si = FP_OFF(TSRNAME)
        intr(0x2F, &reg)
        if (reg.r_ax == 0xKKKK)
                return 1
        return 0
}

//
// Zakaci nase vektore
//
void hook_vectors ()
{
        struct REGPACK reg

        reg.r_ax = 0x3400                  // InDos flag pointer
        intr(0x21, &reg)
        indos = (char far *)MK_FP (reg.r_es, reg.r_bx)
        reg.r_ax = 0x5D06                  // Critical Error flag pointer
        intr(0x21, &reg),
        critical = (char far*)MK_FP(reg.r_ds, reg.r_si),

        reg.r_ax = 0x2100                  // nadji nas DTA
        intr(0x21, &reg)
        dtaseg = reg.r_es                  // sacuvaj ga za posle
        dtaofs = reg.r_bx

        old_8  = getvect (0x8)             // sad sacuvaj stare prekide
        old_9  = getvect (0x9)
        old_13 = getvect (0x13)
        old_2F = getvect (0x2F)
        old_28 = getvect (0x28)

        setvect (0x8, new_8)               // postavi nove
        setvect (0x9, new_9)
        setvect (0x13, new_13)
        setvect (0x2F, new_2F)
        setvect (0x28, new_28)

        // TSR je sada aktivan u smislu prekida
        // jos je ostalo da sam modul savrsi sa radom i od ga odpo
        // izbrise iz memorije ostavljajuci ostatak od vise
}

//
// Izbaci ako mozes TSR iz memorije
//
int unload (void)
{
        unsigned int tsrpsp, envpsp,
        struct REGPACK reg

        if (!already_there())
                return 1

        reg.r_ax = (MY_ID<<8)|UNLOAD_CMD        // neka TSR otkaci prekide
        reg.r_es = FP_SEG(TSRNAME)
        reg.r_si = FP_OFF(TSRNAME)
        intr(0x21, &reg)

        if (reg.r_ax != 0xFFFF)                 // ako nije uspeo
                return CANT_UNLOAD,             // odustani

        // kreso tekogan ... podigni ruke i vrati sve do kraja

        tsrpsp = reg.r_bx                       // ako je uspeo vratili smo
                                                // njegov PSP
        envseg = *(unsigned far*)(MK_FP(tsrpsp, 0x2C)

        reg.r_es = tsrpsp                       // brisemo telo iz memorije
        reg.r_ax = 0x4900
        intr (0x21, &reg)
        if (reg.r_flags&1 != 0)
                return MEM_FROZEN,

        reg.r_es = envseg,                      // brisemo okolinu
        reg.r_ax = 0x4900,
        intr(0x21, &reg)
        if (reg.r_flags&1 !=0)
                return ENV_FROZEN

        return(0)                               // uspeh
}

//
// MAIN
//
int main (void)
{
        unsigned int hotkey

        if (already_there ())           // ako je aktivan deaktiviraj
        {
                int code,

                code=unload()
                puts(message[code]),
                return code
        }

        tick_max=0                      // nemamo aktiviranja tajmerom
        hotkey=ALT+CTRL+SYS_REQ         // odabiramo aktiviranje tasterom

        hotshift = (hotkey >> 8) & 0xFF,
        hotmake = hotkey & 0xFF

        if (hotmake >= K1 && hotmake <= K10)
        {
                if ((hotkey & ALT) != 0)
                        hotmake += ALT_K
                else
                        if ((hotkey & CTRL) != 0)
                                hotmake += CTRL_F
                        else
                                if ((hotkey & (LSHIFT | RSHIFT)) != 0)
                                        hotmake += SHIFT_F
        }

        hook_vectors ()                 // zakacimo vektore

        puts( TSR is activated  )

        keep(0, (_SS  _psp) + 1)        // neka ostane u memoriji sve,
                                        // osim ovog modula
        return 0,
```

Main funkcija i inicijalizacija rezidentnog programa

tačno znamo granicu između potrebnog i nepotrebnog koda i pomoću pomanute funkcije **Keep** (31h prekida 21h) ili neke slične odbacili onaj nepotrebni deo. Međutim, kod C programa je mnogo komplikovanije jer ne znamo kako i gde je pojedini deo koda smešten.

Pogledajmo zato sliku 1, na kojoj se nalazi raspored segmenata u **large** modelu (od sada govorimo o Borlandovom C). U tekstu „Memorija i modeli" se kaže da u tom modelu svaki modul ima svoj segment za kod. Bilo bi, evo ideja, vrlo interesantno da programski kod ovog modula za inicijalizaciju postavimo negde gde ga tačno možemo identifikovati. Još jedan pogled na sliku 1 i imamo rešenje: postavićemo ga tačno na granicu stek-segmenta, tako da nam njegov početak bude raper za oslobađanje memorije (registar SS).

Za ovo ćemo upotrebiti osobenost Borlandovog C-a, odnosno imenovaćemo novi *code* segment za ovaj naš modul. Istovremeno to ćemo učiniti i sa *data* segmentom modula, tako da će ceo modul konačno ispariti kada uradimo ono **Keep**, jer ćemo veličinu programa računati da ponovimo od početka celog programa (što je segment PSP-a) do stek segmenta (što daje SS registar). Dakle, da rezimiramo, sav kod koji je potreban za inicijalizaciju ćemo strpati u jedan modul, a njega zatim izbaciti iz memorije kada nam nije više potreban. U drugi modul ćemo zato staviti sva preuzete prekide, a u treći samu rutinu koja će se pozivati (ova dva poslednja smo razdvojili više iz kozmetičkih razloga). Konačan program nastaje linkovanjem sva tri modula zajedno.

VEKTORI

Prvi modul za koji ćemo se pobrinuti je onaj u kojem ća biti smešteni svi preuzeti prekidi (vektori). Pre svega preuzimamo prekide 08h, 09h, 13h i 28h, o čemu smo već govorili. Pošto je reč o šablonu, moramo omogućiti buđenje TSR bilo nekom kombinacijom tastera, bilo korišćenjem tajmera. U tu svrhu koristimo odgovarajuće promenljive: za tajmer će **timer count** pokazivati broj otkucanih klikova (18,2 puta u sekundi se uvećava) do sada, a **timer_max** broj klikova posle kojih se budi TSR. Ukoliko je **timer_max** nula, naš rezidentni program se uopšte ne treba buditi po tajmeru.

Za kombinacije tastera će biti odgovorni **hotmake** koji će sadržati „make" kod tastera (kod koji se emituje kada se taster pritisne) i **hotshift**, koji sadrži kombinaciju **shift** tastera (**alt, ctrl, shift**). Ukoliko su oba nule, onda se TSR ne budi tasterima. Aktivni deo programa će moći da se pozove na dva načina: iz prekida 08h, kada je dostignut **tick_max** (ako nije bio nula) ili je prethodno bio pritisnut taster (što se beleži u **want_inside**), ili iz prekida 28h ako je prethodno pritisnut taster. Dodatni uslovi se takođe proveravaju: tj. da li je u toku kritični disk proces (**disk_active**), da li je TSR već aktiviran (**im_inside**) i da li je nastupila kritična greška (**critical**). Pre samog ulaza u izvršnu rutinu prebacićemo se na poseban stek, sačuvati PSP segment i DTA i preuzeti prekide koji se brinu o kritičnim greškama i **CTRL-Break** događajima (šta ako se u toku rada, kao njegova posledica, javi kritična greška?).

Preuzećemo još jedan manje poznat prekid, 2Fh, znan kao **Multiplex** prekid. Njegova uloga je malo čudna: on je pre svega namenjen registrovanju aktivnih rezidentnih programa, ali i komunikaciji sa njima, pa i njihovom deaktiviranju. U našem slučaju koristićemo ovaj prekid za tri stvari: proveru da li je naš TSR već u memoriji (da ne bi više kopije bilo aktivno istovremeno), da ga deaktiviramo i da mu prenosemo poruku. U sva tri slučaja, parametri su sledeći: registar AL će sadržati broj funkcije, AH identifikacioni broj našeg TSR, par ES:SI pokazivač na string za identifikaciju, a CX i DX dodatne parametre. Funkcija 00H (AL=00h) proverava postojanje TSR u memoriji: ukoliko je AL jednako identifikacionom broju rezidentnog programa kojem je stigao upit i ukoliko je string na koji pokazuju ES:SI identičan onom koji je zapisan u telu programa, onda rezidentni program vraća AX=FFFFh. Funkcija 01h će takođe izvršiti identifikaciju, ali će zatim pokušati i da vrati sve preuzete prekide na staro stanje, čime se TSR zapravo deaktivira. Ukoliko proces uspe, opet se vraća AX=FFFFh. Funkcije 02h do 80h su za sada nedefinisane, a ove iznad 80h su namenjene komunikaciji. Rezidentnom programu će se za ove iznad 80h slati već opisani kodovi za identifikaciju, a u CX i DX posebni parametri koje će TSR koristiti za neke svoje potrebe (što određuje autor). Uspešan povratak bi trebalo da bude označen sa AX=FFFFh.

Kada se korišćenjem funkcije 01h prekida 2Fh TSR deaktivira, moramo ga još i ukloniti iz memorije. To ćemo obezbediti tako što će TSR po deaktiviranju prekida vratiti u BX svoj segment (segment PSP-a), da bismo mogli standardnom DOS funkcijom taj blok da „obrišemo". Naravno, brisanje će se vršiti iz onog modula za inicijalizaciju, jer brisanje TSR iz memorije vršimo logično sa komandne linije, a ne iz njega samog. Istovremeno moraćemo da se oslobodimo i okoline programa koja takođe ostaje rezidentna sa njim, a njen segment nalazimo iz PSP rezidentnog programa, što je demonstrirano u modulu za inicijalizaciju.

INICIJALIZACIJA

O modulu za inicijalizaciju smo dovoljno rekli. Njegova uloga je dvostruka: ako TSR nije već u memoriji, postavljamo ga tamo, a sam modul pri aktiviranju eliminišemo. Ukoliko jeste u memoriji (što, da ponovimo, proveravamo pozivom prekidu 2Fh), ne aktiviramo još jednu kopiju, već o tom konfliktu obaveštavamo korisnika. Ukoliko želimo brisanje aktivnog TSR iz memorija, šaljemo zahtev za deaktiviranje preuzetih prekida rezidentnom programu prekidom 2Fh, a ovaj nam vraća podatke o svom položaju. Na osnovu toga ga i fizički eliminišemo. I to je sve.

EPITAF

Priloženi tekst programa, podeljen u četiri celine (tri modula i jedno zaglavlje) bi sada trebalo spojiti u jedan projekat i prevesti u large modelu. Telo rezidentnog programa se smešta u funkciju **tsr(unsigned int)** u modul TSR.C, kao i sve ostale potrebne mu funkcije. U ovom istom modulu funkcija **comunicate(int, int, int)** služi za komunikaciju sa onim koji prekidom 2Fh funkcijama 80h i višim šalje neki zahtev programu.

Inicijalizaciju smo implementirali najjednostavnije, odnosno tako da vrši samo dve funkcije: ako je TSR već aktivan – deaktivira ga, ako nije – onda ga aktivira. Može to i mnogo lepše, ali prostor u časopisu nam to nije dozvolio.

Konačno, nikakav primer nismo dali iz prostog razloga što smo smatrali da je već dovoljno rečeno i da je ostatak samo vežba.

TRENDOVI

Telefonija

Nazovi me u osam... pisijem

Prvobitno je deskop računar bio lično sredstvo za rad, najpre sa programima za tabelarne proračune ili tekst procesorima, a kasnije sa drugim uslužnim alatkama, kao što su baze podataka, menadžeri projekata, prezentacioni programi. Onda su lokalne mreže (LAN) povezale desktop računare u jednoj radnoj grupi ili odeljenju, čime smo dobili mogućnost da koristimo zajedničke informacije i resurse i da komuniciramo elektronskim putem.

Sledeći korak u evoluciji desktop računara je kombinovanje računara sa uređajem bez kojeg se poslovni, a ni život uopšte, odavno ne može zamisliti – sa telefonom. „Intel" i „Microsoft" su u saradnji sa više velikih telefonskih, računarskih i softverskih kompanija kreirali standardni interfejs – *Windows Telephony Application Programming Interface* – za integraciju telefona i personalnog računara (koji radi pod „Microsoft"-ovim operativnim sistemom). Mogućnosti koje se time dobijaju su fantastične, kako za korisnike telefona tako i za korisnike računara. Tek će *Windows Telephony API* učiniti da dođu do izražaja neke postojeće, ali nedovoljno iskorišćene mogućnosti telefona. Na primer, korisnici će lako pristupati svim telefonskim funkcijama preko *Windows* grafičkog interfejsa. Proizvođači telefona odavno nude prosleđivanje poziva na drugi telefon, brzo biranje brojeva, kao i tzv. konferencione pozive, ali većina ljudi koristi samo dve-tri osnovne funkcije. Računarski softver će im sada omogućiti da s lakoćom koriste čak i one najkompleksnije funkcije – na primer, konferencijski poziv se obavlja jednim klikom na imena učesnika konferencije.

Uz grafički interfejs možete prepustiti svom računaru da prima telefonske pozive. Nećete morati da brinete što niste kod kuće u vreme kada očekujete važan poziv – kompjuter će ga automatski prosledit do mesta na kojem očekujete da ćete u to vreme biti. Osim toga, obična telefonska slušalica se u ovakvoj kombinaciji može koristiti kao uređaj za glasovni *input/output*. Tako ćete nekom izveštaju moći da dodate usmene komentare ili da slušate glasovne poruke uz ono što se prikazuje na monitoru.

Samo se po sebi razume koliko će ovakvo direktno povezivanje postojećih aplikacija sa telefonskom mrežom doprineti njihovoj upotrebnoj vrednosti, kao i produktivnosti korisnika. Tako će, na primer, program za upravljanje ličnim informacijama moći ne samo da potraži neki telefonski broj, već i da ga nazove.

Kombinovanje računara i telefona će omogućiti i rad nekoliko zanimljivih novih aplikacija, kao što je, na primer, integrisano slanje i primanje poruka. Naime, tri oblika poruke – glasovne, elektronske i faks-poruke – će moći da se pregledaju na jednom mestu na računaru. Osim toga, ljudi na geografski udaljenim mestima će moći da održavaju sastanke preko telefona, imajući na svojim računarima iste vizuelne i zvučne informacije.

Jedan od najvećih problema pri povezivanju telefona i kompjutera čini mnoštvo raznih telefonskih uređaja i opreme i nekompatibilnost protokola, koji se razlikuju od mreže do mreže. *Windows Telephony API* je standardni API koji korisnicima i proizvođačima softvera omogućava da iskoriste sve što telefon nudi, oslobađajući ih brige o načinu rada telefonske mreže na koju su priključeni. Drugim rečima, API je potpuno nezavisan od hardvera, načina na koji je ostvarena veza između telefona i računara, što programerima daje potpunu slobodu u pisanju programa, a korisnicima fleksibilnost u pogledu izbora konfiguracije.

Windows Telephony API je deo „Microsoft"-ovog koncepta zvanog WOSA – *Windows Open Services Architecture*, koji nudi otvoren skup interfejsa za pisanje aplikacija i za pristup aplikacijama. *Windows Telephony* projekat je za sada usmeren na desktop računare, dok će sledeća izdanja biti proširena i na serverska okruženja. „Intel" i „Microsoft" pozivaju na saradnju druge proizvođače sistemskog softvera, želeći da njihova nova specifikacija dobije primenu i na drugim platformama. U međuvremenu, ona je već dobila podršku od četrdesetak kompanija – proizvođača telefonskih centrala, računara i perifernih uređaja, softvera i mrežne opreme.

Mala izmena – veliki problem

U ovoj seriji sam već pisao o funkciji **time** koja vraća proteklo vreme u sekundama od nekog, u funkciji definisanog "nultog" dana. Taj početni datum za upotrebu funkcije i nije mnogo bitan – broj proteklih sekundi od, recimo, 1. januara 1980. retko da nekoga zanima, ali se zato ova vrednost dalje koristi u drugim funkcijama koje mogu da je "rastave" na godinu, mesec, dan, sat i minut, pa odatle dalje da ispišu datum u tekstualnom obliku i slično. I naravno, oduzimanjem dve *time_t* vrednosti (**time_t** je tip za vreme, praktično **long**) lako izračunavamo broj proteklih dana između dva datuma, ili merimo duže vremenske intervale. Dakle, početni datum može biti interesantan u dva slučaja – kada želimo da binarni sadržaj rezultata prenesemo u neko drugo okruženje ili kada **time_t** vrednosti koristimo za pamćenje datuma, jer smo tada ograničeni na određeni period.

Listajući dokumentaciju pod naslovom "Šta je novo u verziji MSC 7.0", primetio sam da je početni datum pomeren sa ranije vrednosti 1. januar 1980. na 1. januar 1900. godine. Preciznije, funkcija time sada vraća broj sekundi proteklih od ponoći 31. decembra 1899. godine. **time_t** tip je i dalje ostao u **long** formatu, ali je promenjen u **unsigned**, jer su za ove naše dane vrednosti već dovoljno velike da je potreban pun neoznačeni opseg koji long promenljiva može da drži. To konkretno znači da će pre 6. februara (sreda) 2036. godine i to pre 6:28:14 sati morati da izađe bar još jedna verzija MSC kompajlera, jer tačno tada ističe opseg koji je moguće prikazati **unsigned long** promenljivom ako se zadrži isti početni datum. Ova promena mi se tada učinila prilično beznačajnom i, ako se dobro sećam, pravdana je usaglašavanjem sa standardom, tako da sam brzo na nju zaboravio.

Prošlog meseca je preda mnom bio zadatak od kojeg inače pokušavam da pobegnem kad god mogu – trebalo je analizirati izvorni kod nekog programa i videti da li se program može prilagoditi mojim potrebama. Na sreću, pokazalo se da je firma koja je program pisala uradila posao sasvim dobro, tako da se izmena svela na par trivijalnih intervencija. Ipak, pojavio se jedan problem.

Procedura prevođenja je potpuno automatizovana **makefile** datotekom, a svi moduli su pažljivo pisani uz maksimalno vođenje računa o prenosivosti. Aktiviranjem odgovarajuće **makefile** datoteke, ceo paket se može prevesti i sa nama najinteresantnijim kompajlerima, *Borland TC* i *Microsoft C 6.0*. Moja opcija je bila *Microsft C 7.0* koji isključivo koristim u poslednje vreme, ali sam pretpostavio da probleme sa prevođenjem neće biti.

I zaista, ceo paket je preveden bez i jedne greške, ali se na dosta mesta pojavilo upozorenje "Conditional expression is constant". Ovakvo upozorenje retko izaziva zabrinutost – mnogi autori vole da napišu

```
while ( 1 )
```

u situacijama kada izlazak iz petlje nema dominantan uslov već ih ima više u telu petlje. Druga mogućnost za pojavu ovakvog upozorenja je najčešće omaška u pisanju, recimo

```
while ( t = 1 )
```

kada je znak poređenja == greškom skraćen na znak dodele. Ipak, sudeći po stilu pisanja i ozbiljnosti celog paketa, obe ove situacije su teško mogle biti uzrok pojava upozorenja. Pregled mesta na kojim se upozorenje javlja je otkrio da se uvek radi o jednom te istom kodu poput ovoga

```
time_t limit,ostaje,
ostaje = limit – time( NULL ),
if ( ostaje < 0 )
{ ...
```

Ideja je dakle da se proveri da li je zadato vreme nekog procesa isteklo i da li program treba da odustane od daljeg izvršavanja. Problem je naravno u tome što je u MSC 7.0 **time_t unsigned** tip, pa otuda uslov (**period < 0**) nikada ne može biti ispunjen. Striktno gledano, autor programa je kriv za ovu grešku, jer se oslonio na činjenicu da iza tipa **time_t** stoji tip **long**, što svakako nije dobra praksa. Ipak, teško da ovakva greška može ići njemu na dušu. To što postoji neki poseban tip **time_t** za vreme, u praksi gotovo ništa ne znači, jer smo prinuđeni da ga tretiramo kao **long**. Za merenje razlike između dva vremena ne postoji funkcija, pa ostaje da koristimo oduzimanje i tu cela teorija o posebnom tipu pada u vodu (šta ako se u nekom trenutku javi potreba da **time_t** bude struktura?). Doduše, funkcija **difftime(time_t, time_t)** postoji u MSC biblioteci, ali ona vraća **float** rezultat – to znači da ćete u program uvući i priličan deo biblioteke koji vam uopšte nije potreban, pa je pitanje koliko je ova funkcija primenljiva.

Rešenje u konkretnom slučaju je jednostavno, dovoljno je ispitivanje izvršiti drugačije

```
if ( time( NULL ) > limit )
{ ...
```

Ipak, ako nam je potreban podatak o vremenu koje proces još ima na raspolaganju (promenljiva "ostaje"), onda se stvar nešto komplikuje. Moguće rešenje je

```
time_t limit,ostaje,t,

time( &t ),
ostaje = t > limit ? 0 : limit – t,
if ( ostaje == 0 )
{ ...
```

Dakle, ne samo da je potrebno na puno mesta dodati jednu pomoćnu promenljivu, već treba pažljivo na svim mestima poređenje "**ostaje < 0**" zameniti sa "**ostaje == 0**", u šta treba uračunati i eventualne funkcije kojima se ovaj podatak prosleđuje kao argument. Sve zajedno dovoljno posla da se izgledima da se poneko mesto propusti, što bi bilo katastrofalno po rad programa, koji bi u takvoj petlji ostao večno.

U toku probe pri pripremi ovog teksta isplivao je još jedan interesantan detalj. Naime, da bih proverio šta se tačno dešava, izdvojio sam samo kritičan deo i preveo ga – upozorenje se nije pojavilo! Ova proba je prevedena bez ikakvih optimizacija, pa je bilo očigledno, pošto je to jedina bitna razlika u odnosu na original, da se tu krije razlog. I zaista, uključivanjem bilo koje od optimizacija, kompajler je detektovao grešku.

Problem je konačno rešen na najjednostavniji način – ponovnim instaliranjem MSC 6.0. Ipak, priča više nego dobro ilustruje koliko je pitanje prenosivosti osetljiva stvar i koliko u C-u, koji se tom osobinom hvali, još uvek ima nedoslednosti koje je mogu lako narušiti. Konačno, jedan dobar praktičan primer koji govori u prilog objektnih jezika.

Pascal je brži

Brzine, uz što manju potrošnju još uvek dragocene RAM memorije, nikada dosta. Pri tome se ovaj zahtev može ispuniti na dva načina. Prvi, i to onaj koji jedini može dati značajne pa i spektakularne rezultate, je rešavanje problema na algoritamskom nivou. Drugi metod, koji će nadamo se svi uskoro nestati iz programerske prakse i biti potpuno prepušten kompajleru, su razne tehnike koje se oslanjaju na poznavanje mikroprocesora ili ukupno posmatrano, hardvera računara. Ove druge nikada ne daju spektakularne rezultate, ali su u uslovima ograničenih resursa ponekad dovoljne da se napravi zadovoljavajući korak. Ako tehnika ne zahteva nikakvu izmenu programa, tim bolje – nema razloga da je ne probamo. Jedna od takvih retkih bezbolnih tehnika je korišćenje *pascal* konvencije za prenos argumenata funkciji.

Jedna od osobina C-a koja se ređe sreće kod drugih programskih jezika je mogućnost prenosa promenljivog broja parametara funkciji. Ona se ostvaruje tako što se argumenti funkcije na stek postavljaju obrnutim redom od onoga kojim su u funkciji deklarisani (dakle, sa desna na levo). U trenutku poziva funkcije na stek se gura još jedan parametar – povratna adresa na koju se treba vratiti nakon što funkcija završi rad. Ova adresa služi kao neka vrsta sidra – to je zapravo jedina tačka na steku za koju funkcija može da zna. Pošto je pre nje na stek postavljen i poslednji parametar redosledom sa desne na levo, funkcija na steku vidi prvo povratnu adresu, a potom redom parametre, i to od prvog ka poslednjem, njihovim prirodnim redosledu – onako kako smo ih i deklarisali. Funkcija zato može da na steku bez problema adresira parametre ovim redom, jedino što ne može je da odredi koliko ih ima, pa se zato u ovakvim funkcijama uvodi dodatni parametar koji upisuje šta je sve na stek stavljeno (kao kod **printf** funkcija) ili se pak uvodi marker argument – poslednji argument u listi obično ima neku zgodnu vrednost, najčešće NULL.

Mala nevolja ovog sistema je to što sama funkcija ne može da skine parametre sa steka, već to mora da se obavi iza mesta poziva funkcije, jer samo program koji je funkciju pozvao tačno zna šta joj je na stek postavio. Tako se poziv C-ovske funkcije

```
funkcija(par1, par2, par3),
```

na 80x86 procesorima prevodi kao

```
push  par3
push  par2
push  par1
call  funkcija
add   sp, sizeof( par1) + sizeof( par2 ) + ...
```

Sve ovo izgleda lepo i fleksibilno, da nije jedne sitnice – Intelovi mikroprocesori poseduju i mogućnost da se RET instrukciji mikroprocesora navede i opcioni parametar – broj bajtova za koje treba uvećati stek pointer nakon što se skine povratna adresa. Konkretno, umesto instrukcije **add sp**... na kraju prethodnog primera, moguće je u samoj funkciji izvesti RET 6 (ako su **par1..3** int-dužine). Doduše, tada sama funkcija mora da zna tačno šta je zalegla na steku, pa je onda ovaj problem najjednostavnije rešiti tako što se zabrani prenos promenljivog broja parametara i tada se parametri mogu na stek postavljati redom kojim su deklarisani. Tako stižemo do onoga što se naziva *pascal* konvencija za prenos parametara.

Pascal konvencija je, sa stanovišta mikroprocesora, efikasnija – nešto je kraća (RET pa ADD SP,x je dugačko 4 bajta, a RET x zauzima 3), nešto brža (RET i RET x se praktično ne razlikuju u brzini, a u drugom slučaju se štedi vreme potrebno za dodatni ADD SP,x). Iz ove računice proizlazi neznatan dobitak u brzini, ali je u praksi ponekad za nijansu veći. Naime, kompajleri često optimizuju skidanje parametara sa steka POP instrukcijom u slobodan registar, umesto sabiranja vrednosti na SP, pa je u tim slučajevima dobitak nešto veći.

Osnovno pitanje koje verovatno sebi postavljate je da li sve to ima ikakvog smisla. U nekoj drugoj situaciji odgovor bi bio da zbog par "bajtova i nešto klokova" zaista nema razloga da ulažemo bilo kakav napor. Ova situacija je ipak specifična, jer nikakav poseban napor i nije potreban – dovoljno je pri prevođenju navesti /Gc prekidač, čime se sve funkcije prevode u *pascal* konvenciju. Praksa pokazuje da dobici mogu biti do 10% ukupne dužine programa (recimo, program sa 150K može pasti na 135k, što ipak nije beznačajno), dok je brzinu nešto teže proceniti na ovako globalnom nivou. Bolje prolaze oni programi koji su sastavljeni iz većeg broja funkcija, a posebno oni koji sadrže neku vrstu seta baznih fukcija koje se potom masovno koriste u ostatku programa.

Treba imati na umu da nas ovaj prekidač ne ostavlja bez mogućnosti pisanja funkcije sa promenljivim brojem parametara. Dovoljno je u deklaraciji funkcije navesti

```
_cdecl funkcija(par1, ...)
```

i tako lokalno za funkciju ostaviti C konvenciju sa svim što uz to ide. Moguće je i obratno – koristiti inicijalnu C konvenciju, a samo pojedine funkcije deklarisati kao *pascal*

```
_pascal funkcija(par1...)
```

Za one koji mešaju C sa asemblerom, potrebna je još samo napomena da se nazivi funkcija po *pascal* konvenciji prevode u velika slova i nemaju nikakve dodatne prefikse i sufikse – dakle, funkcija postaje FUNKCIJA za razliku od C-ovskog imena _funkcija.

MT COMPUTERS

Preduzeće „MZ" 11000 Beograd
Dubljanska br. 70 Tel. 011/434 812
Fax 011 450 471

ŠTAMPAČI EPSON

LX-400 9-pin A4 format 180 z/s čir/lat
LX-100 9-pin A4 format, 240 z/s CSF YU-set
LX-850 9-pin A4 format 200 z/s YU-set
FX-870 9-pin A4 format 380 z/s YU-set
FX-1050 9-pin A3 format 200 z/s YU-set
FX-1050 9-pin A3 format 300 z/s čir/lat
LQ-100, 24-pin A4 format 200 z/s CSF čir/lat ESC P 2
LQ-570, 24-pin A4 format 240 z/s YU-set ESC P-2
LQ-870 24-pin A4 format 330 z/s YU-set ESC P 2
LQ-680, 24-pin A4 format 300 z/s YU-set kolor
LQ-1070 24-pin A3 format 240 z/s YU-set ESC P-2
LQ-1170 24 pin A3 format, 330 z/s YU-set ESC P 2
LQ-1060 24-pin A3 format 300 z/s YU-set kolor
DLQ-2000 24-pin A3 format 270 z/s YU-set kolor
DFX 5000 9-pin A3 format, 533 z/s YU set
DFX-8000 18-pin A3 format 1066 z/s YU-set
SQ-870 48-mlaz A4 format 660 z/s YU set INK JET
SQ-1170 48-mlaz A3 format 660 z/s YU set INK JET
EPL-4000 LASER 300 DPI 6 s/m 1 MB H-YU-set (opcija)
EPL-4300 LASER 300 DPI 6 s/m 1 MB RITech H-YU-set (opcija)
EPL-8100 LASER 300 DPI 10 s/m 2 MB RIT PCL 5 H-YU set (opcija)
EPL 7500 POSTSCRIPT LASER MEGA2 300 dpi 6 s/m 2 MB
GT-8000 COLOR IMAGE SCANER A4 format 800 DPI

Roland
DIGITAL GROUP

DXY 1100 A3 format 8 pera 42 cm/s mag držanje papira
DXY 1200 A3 format 8 pera 42 cm/s ele statičko papira
DXY 1300 1MB A3 format 8 pera 42 cm/s ele statičko papira
SKETCH MATE A3 format 8 pera
SKETCH MATE A4 format 8 pera
DPX 2500 A2 ploter tabla 8 pera, 62 cm/s 1 MB
DPX 3500 A1 ploter tabla 8 pera 62 cm/s 1 MB
GSX-3000 A1 „ROLL" ploter 8 pera 113 cm/s 1 MB aut p v p
GRX 300 AG A1 „ROLL"-ploter 8 pera 66 cm/s
DPX-4600 A0 ploter tabla 8 pera 62 cm/s 1 MB
GSX-4000 A0 „ROLL"-ploter 8 pera 113 cm/s 1 MB aut p v p
GRX 400 AG A0 „ROLL" ploter 8 pera 66 cm/s
LTX 100 A3 format termički ploter
LTX 120 A3 format termički ploter
LTX 321 A1 format termički ploter
LTX-420 A0 format termički ploter
Data buffer SYA 550
CAMM 1 PNC 1100 PLOTER REZAČ format od 50 do 640 mm (do 1600 mm)
CAMM 1 PNC-1800 PLOTER REZAČ A0 format (do 3600 mm)
STIKA Rezač sa ugrađenim skenerom sa RS232C i programom

EIZO

EIZO 9080i 16" kolor monitor 1024x768
EIZO 6500 21" monohromatski monitor 1664x1200 (60Hz) 1280x1024
EIZO F550i 17" kolor mon ravan ekran max 1280x1024
EIZO F750i 21" kolor mon ravan ekran max 1280x1024
EIZO T560i 17" kolor (TRINITRON CRT) 1280x1024 CRT Tno p 0.31
EIZO T600 20" kolor (TRINITRON CRT) 1280x1024
EIZO GRAFIČKE KARTICE — VA30 VA41 AA51 MD B09 MD B10 MD B12

GARANCIJA GODINU DANA
POSEDUJEMO DODATNI I POTROŠNI MATERIJAL
ĆIRILIČNO-LATINIČNI EPROMI ZA LX FX,LQ
MODELE OVLAŠĆEN SERVIS
INFORMACIJE NA TEL 011 434 812 FAX 011 450 471

386 SX	33 MHz		2 MB	1,490.-
386 DX	40 MHz		4 MB	1,790.-
486 DX	33 MHz		4 MB	2,490.-
486 DX	50 MHz		4 MB	2,920.-
486 DX	50 MHz	EISA	4 MB	3,450.-

STANDARD ZA KONFIGURACIJU

- tvrdi disk QUANTUM 85 MB 16 ms
- flopi disk TEAC 5,25 "
- kombinovani kontroler (2S, 2P, 1G)
- SVGA monohromatski monitor
- TRIDENT 9000
- Mini tower
- CHICONY tastatura

DOPLATA ZA

1 MB memory SIMM/SIPP	70,-
TRIDENT SVGA 8900 1 MB	50,-
OAK 087 1 MB	50,-
TSENG ET4000 VESA 1 MB	50,-
TSENG ET4000 enhanced 32k boja	250,-
S3 windows akcelerator	300,-
ATI ULTRA sa koprocesorom i mišem	650,-
floppy disk 3½" 1 44 MB	120,-
hard disk MAXTOR 120 MB 16 ms	100,-
hard disk CONNER 170 MB 16 ms	150,-
hard disk MAXTOR 210 MB 13 ms	350,-
hard disk QUANTUM 250 MB 13 ms	400,-
hard disk QUANTUM 550 MB 13 ms	1550,-
GALAXY IDE cache controller	400,-
midi tower 200 W	30,-
slim line	70,-
color monitor DATASTAR 1024x768	400,-
PHILIPS BRILLIANCE 7CM3209	600,-

SPECIJALNA PONUDA
600,-
BESPREKIDNO NAPAJANJE
UPS 550 VA

IZBOR IZ PONUDE

3M diskete 5,25" MD-2HD	25,-
3M diskete 3,50" FD-2HD	35,-
štampač EPSON LX-400, 9 pin, A4	480,-
štampač EPSON FX-1050, 9 pin, A3	1200,-
štampač EPSON LQ-1070, 24 pin, A3	1300,-
štampač EPSON LQ-570, 24 pin, A4	800,-
štampač IBM proprinter, 24 pin, A4	850,-
štampač HP IIIP sa tonerom	2600,-
štampač HP IV sa tonerom	3800,-
traka za LX-400	15,-
toner za IIP/IIIP	200,-
Miš GM-One	50,-
Miš GM-D320	60,-
Miš GM-6000	70,-
memorija za IIP/IIIP	300,-
Mobile rack za 3½" hard disk	150,-
Monitor filter stakleni	50,-
Podloga za miša	10,-
Skener GS-45000	300,-
Džojstik MS 500	40,-
Grafička kartica ATI ULTRA	700,-
Grafička kartica S3 Windows	400,-
koprocesor 287-20 MHz IIT	100,-
koprocesor 387-20 MHz INTEL	150,-
koprocesor 387-40 MHz IIT	200,-
fax/modem interni AMIGO	250,-
fax/modem externi DISCOVERY	500,-
modem interni CARDINAL	160,-
modem externi DISCOVERY	380,-

ADACOM
C O M P U T E R S

CLIPPER SAVETNIK (7)

Nenad Batočanin

Slovi u Clipper-u

U *Clipper* programima se obično radi sa veoma raznorodnim podacima. Iako bi postojanje slogova kao što je **RECORD** u *Pascal*-u ili **struct** iz *C*-a bilo veoma logična i korisna stvar. Loša vest je da autori *Clipper*-a ne daju ni nagoveštaje o dodavanju pravih slogova u *Clipper* (mada bi se objekti mogli posmatrati i kao slogovi). Dobra vest je da se slogovi mogu na izvestan način zameniti nizovima, zahvaljujući osobini *Clipper*-a da nizovi mogu imati elemente proizvoljnog tipa. Na primer, slog koji bi sadržao podatke za jednog studenta mogao bi izgledati ovako:

aStud := { "Pera Peric", 1990, 3}

Za nekog sa strane ovo bi teško moglo da se protumači kao slog koji se odnosi na studenta koji je upisao treći semestar, a fakultet upisao 1990. godine. Izrazi tipa **aStud[2]** u programu ne znače previše. Možda bi konstrukcija poput ova malo pomogla:

aStud := {cIme = "Pera Peric", nGod = 1990, nSem = 3}

Ovde se koristi *in line* dodeljivanje za istovremeno zadavanje članova niza i inicijalizaciju memorijskih promenljivih. Može se koristiti u slučaju da postoji samo jedan niz **aStud** – čim se broj ovakvih slogova poveća, rešenje postaje nepriklađno. Notacija poput **aStud.cIme** je još uvek nemoguća. Znači trebalo bi redni broj elementa niza opisati na neki „humaniji" način od rednog broja. Jedno od zadovoljavajućih rešenja nudi pretprocesor. Prvo se definišu konstante za svaku poziciju u nizu:

```
#define IME 1
#define GOD 2
#define SEM 3
```

Sada je ime studenta dato sa **aStud[IME]**, godina upisa sa **aStud[GOD]** i slično. Naravno, ovo je daleko preglednije od prethodnih rešenja i uvek treba nastojati da se u programu koristi ovakva notacija. Heder datoteke koje se dobijaju uz *Clipper* obilato koriste ovakav metod za pretvaranje nerazumljivih brojeva u simboličke konstante: svakom više znači **K_ALT_N** nego 305!

Precizno gledajući, u *Clipper*-u postoje slogovi, to su slogovi u **DBF** datotekama. Ovi slogovi se učitavaju u memoriju samo interno, ali se to može ispraviti odgovarajućim procedurama. Sledeća procedura učitava slog iz datoteke u memorijski niz:

```
aBuffer := {}
FOR i := 1 TO FCount()
  AAdd (aBuffer, &(FieldName(i)))
NEXT
```

ovakav niz se jednostavno ponovo upisuje u izabrani slog:

```
FOR i := 1 TO Len(aBuffer)
  FIELD->&(FieldName(i)) := aBuffer[i]
NEXT
```

Ovaj metod se može koristiti za kopiranje jednog sloga datoteke u drugi i sl. Uslov je da oba sloga imaju istu strukturu, ali nije teško postići da radi i za slogove različite strukture.

Kako preneti argumente

Prenos argumenata u procedure i funkcije je sasvim jednostavna stvar: definišete proceduru kao **PROC Test (x, y)**, a zatim je pozivate sa **Test(11, 'ABC')** ili na neki sličan način. Problem kojim ćemo se ovde baviti je izbor i način prenosa argumenata. Na ovo se po pravilu obraća veoma malo pažnje, ali je reč o veoma važnom problemu. Naravno, kod jednostavnijih funkcija je stvar sasvim prosta i verovatno nećete nikad imati većih problema ako ste argumente loše izabrali. Ali, ako je reč o složenoj funkciji koja ima veći broj argumenata i koju ćete veoma često koristiti u više programa, veoma je važno da **dobro** razmislite o koncepciji i budućnosti te funkcije.

Treba imati u vidu da osim standardnog prenosa parametara u *Clipper*-u postoji mogućnost prenosa vrednosti preko globalnih promenljivih (**PUBLIC, PRIVATE, STATIC**), ali je ovo rešenje svakako manje preporučljivo, jer u sebi krije opasnost od bočnog efekta. Tako se može desiti da analizirate neku proceduru i naiđete na korišćenje globalne promenljive, a da niste sigurni u njenu vrednost i značenje. Ovakvim rešenjem se vezujete za procedure prethodnog nivoa. Mada stvaranje takve zavisnosti ponekad bitno može pojednostaviti program, u principu ga treba izbegavati.

Dakle, pravi izbor je argument lista. Na primer, potrebno je napraviti opštu proceduru za tabelarni pregled. To bi moglo da izgleda ovako:

PROC Table(x1,y1,x2,y2,cFile)

Reklo bi se sasvim logično i prirodno: prvo se zadaju koordinate prozora u kome se vrši pregled, a zatim naziv **DBF** datoteke koju treba prikazati. Međutim, potrebe stalno rastu: prilikom korišćenja funkcije **Table** ukazala se potreba da definišete još neki argument – na primer, indikator da li treba ispisati okvir oko pregleda i sl. Naravno, stvar se jednostavno rešava dodavanjem novog parametra. Međutim, tada nastaju problemi: staru verziju funkcije pozivate na puno mesta i svuda treba promeniti (dodati) argumente. Ako se ova operacija uradi jednom ili dva puta, ništa strašno, postoje brojni editori i alatke koje olakšavaju ovaj posao. Međutim, ovakve operacije su siguran znak da funkcija nije dobro koncipirana.

Jedno rešenje bi moglo biti u promenljivom broju argumenata. *Clipper* ne proverava broj stvarnih (pri pozivu) i formalnih (kod definisanja funkcije) parametara. Ako se pošalje više parametara, višak će jednostavno biti ignorisan, a ako se pošalje manje argumenata, ostali će dobiti vrednost NIL, u zavisnosti od metoda prenosa (argument lista ili preko **PARAMETERS** naredbe). Ovakav sistem je idealan za promenljiv broj argumenata i koristi ga dosta *Clipper*-ovih funkcija. Na primer, funkcija **SubStr(cString, nStart, nCount)** služi za izdvajanje podstringa zadatog stringa **cString**, počev od pozicije **nStart**, na narednih **nCount** simbola. Treći parametar nije obavezan i, ako se ne navede, podrazumeva se da treba uzeti string do kraja. Funkcija treba da predvidi koji argumenti su neobavezni i da im dodeljuje neka podrazumevane vrednosti. Na primer, početak funkcije **Table** može izgledati ovako:

```
PROC Table (x1,y1,x2,y2,cFile,nBox)
  IF nBox == NIL
    nBox := 1
  END IF
IF

RETURN
```

Sada se može pozivati sa **Table(0,0,24,79,"TEST",1)** ili sa **Table(0,0,24,79,"TEST")**. Na isti način se može predvideti i izostavljanja parametara u sredini liste. Na primer, funkcija **Table** se može napraviti tako da se, ako se ne navedu koordinate, podrazumeva ceo ekran:

Table(,,,,cFile)

Obrada ovako izostavljenih parametara je potpuno analogna kao u prethodnom primeru.

Ako funkcija ima veliki broj argumenata, pojaviće se problem redosleda argumenata. Za primer pogledajte funkciju **DBEdit**: da li ste ikada zapamtili sve njene argumente? Zato sve složenije funkcije obavezno treba maskirati pretprocesorom i tako napraviti komandu kod koje ključne reči jasno pokazuju koji argument ima koje značenje. Na primer:

@ X1,Y1,X2,Y2 – BROWSE cFile DOUBLE

O tome kako se realizuju ovakve komande bilo je reči u ranijim brojevima "Računara".

Kod veoma složenih funkcija metod prenosa preko argument liste može biti veoma nepriklađan. To se javlja zato što broj parametara može biti veoma veliki i odgovarajuća funkcija više neće biti nimalo elegantna za rad. U tom slučaju se može primeniti prenos argumenata preko niza. Svi srodni argumenti se grupišu u jedan niz i prenos argumenata se svodi na prenos nekoliko nizova. Na primer, za tabelarni pregled se mogu prenositi samo tri parametra: niz u kojem se definišu kolone, niz u kojem se definišu razni opšti parametri i niz operacija koje se aktiviraju pritiskom na određeni taster. Tako bi se poziv veoma komplikovanog tabelarnog pregleda sveo na:

Table (aColumn, aSys, aAction)

Sada nastaje problem snalaženja u odgovarajućim nizovima. Niz koji definiše kolone nije problem, jer svaki element ima sasvim jasno značenje: **aColumn[4]** opisuje četvrtu kolonu tabelarnog prikaza, obično preko kodnog bloka. Međutim, šta sa raznim parametrima smeštenim u **aSys**? Ovde je najbolje primeniti tehniku iz prvog dela: definisanje simboličnih konstanti za svaku poziciju u nizu. Na primer:

```
#define T_BOX        1
#define T_SHADOW     2
#define T_XY         3
...
#define T_HELP      10
#define T_MAX       10 // Broj stavki
```

sada program koji poziva funkciju **Table** izgleda ovako:

```
// Kolone
aColumn := {}
AAdd (aColumn, {...})

// Opcija
aSys := Array(T_MAX)
aSys[T_SHADOW] := .T.
aSys[T_HELP] := "PREGLED PODATAKA"

// Pregled
Table (aColumn, aSys) // Izostavljen je aACTION
```

Ovaj sistem se u praksi pokazuje kao veoma zahvalan: ako vam zatreba neki novi argument, jednostavno ga dodate u neki od tri postojeća niza, a stari programi ostaju potpuno isti.



RISC PITALICA

84. Pitalica se, za razliku od većine ranijih mikroprocesorski orijentisanih problema, pokazala kao prilično tvrd orah – samo sedam od primljenih 35 kupona sadržalo je neophodne *dve* instrukcije!

Podsetimo se pre svega problema koji se „događa" na jednom stilizovanom RISC procesoru. Kao i većina modernih 32-bitnih RISC procesora, „naš" čip poseduje trideset dva 32-bitna registra, koji se obeležavaju sa R0, R1, ... R31. Registar R0 je „uzemljen" tako da, bez obzira na ono što u njega upisujete, uvek ima vrednost 0, čime se postižu izvesne uštede u broju neophodnih instrukcija. Za ovaj zadatak značajne su nam naredbe MOV *odredište, izvor*, ADD *odr, izv*, SUB *odr, izv*, AND *odr, izv*, OR *odr, izv*, XOR *odr, izv*, SLL *odr, koliko*, SLR *odr, koliko* i SAR *odr, koliko*, BEQ *reg1, reg2, odredište* i BGE *reg1, reg2, odredište*. Smisao većine instrukcija je verovatno jasan – MOV je prenošenje sadržaja, ADD sabiranje, SUB oduzimanje (računa se *odredište = odredište-izvor* u potpunom komplementu), AND, OR i XOR konjukcija, disjunkcija i ekskluzivna disjunkcija, SHL i SHR aritmetičko šiftovanje *odredišta* ulevo i udesno za zadati broj bita (pri šiftovanju u registar „ulazi" nula), SAR aritmetičko šiftovanje udesno (u registar „ulazi" bit jednak prethodnom sadržaju najvišeg bita, tj. čuva se znak) dok su BEQ i BGE uslovni skokovi – s obzirom da procesor nema flegove, skok na *odredište* će nastupiti ako je sadržaj prvog navedenog registra jednak (odnosno kod instrukcije BGE veći ili jednak) sadržaju drugog, pri čemu se brojevi posmatraju kao označeni. Mnoge od ovih instrukcija, kao što ćemo videti, nisu neophodne za rešavanje zadatka, ali smo želeli da obezbedimo što više prostora za neka „avangardnija" rešenja.

Set instrukcija je potpuno ortogonalan, tj. svaka se operacija može obaviti nad sadržajem registra ili nad konstantom – ADD R1, R2, na primer, računa R1 = R1+R2, a ADD R1, #3 računa R1 = R1+3 (realan procesor bi, pored svih principa RISC arhitekture, morao da obuhvati i instrukcije za operisanje sa podacima u memoriji, ali one za ovaj problem nisu bitne). Pri radu sa konstantama, međutim, dolazi do izražaja specifičnost RISC tehnologije: da bi se svaka instrukcija sa adresom kodirala u jednu reč, ostavljeno je samo 13 bita za konstantu – dopuštena su, dakle, samo operacije sa konstantama između 00 i 1FFFh. S obzirom da se ove konstante tretiraju kao **označeni** brojevi, prilikom operacija se njima vrši se ekspanzija znaka. Ukoliko na primer izvršimo MOV R1, #1001h, u registru R1 će se zapravo naći broj FFFF0001h, pošto se vodeća jedinica „proširila" na čitav registar. Slično se dešava i kada, na primer, saberemo ili logički množimo R1 sa 1001h – konstanta se najpre interno „proširi" na 32 bita, pa se onda obavi operacija. Slika 1 pokazuje kako se to dešava.

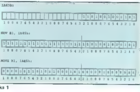
Da bi se u registre lakše smeštali podaci, obezbeđena je i naredba MOVX *registar, konstanta* koja uzima 19 bita konstante (19+13 je tačno 32) i smešta ih u **viših** 19 bita registra, popunjavajući istovremeno nižih 13 bita nulama. Delovanje ove instrukcije takođe je prikazano na slici 1. Napomenuli smo da instrukcija ADDX, ANDX i slične ne postoje – ovakvo prenošenje podataka moguće je samo kod instrukcije MOVX.

Posle opisa procesora, sledi problem. Posmatrajmo kod koji kompajler generiše prevodeći izraz našik na R=0D6D8B627h. Treba, dakle, dovesti broj D6D8B627 u R1 primenjujući, naravno, što manje instrukcija. „Očigledna" ideja da izvršite MOVX R1, #6B6CDh a zatim ADD R1, #1827h na žalost ne daje rezultate jer će se proširenjem vodeće jedinice broja 1827h na 32 bita obaviti sasvim pogrešno sabiranje. Koliko je instrukcija potrebno da bi se bilo koja 32-bitna konstanta (pa i pomenuti broj) upisala u R1?

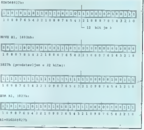
„Očigledno" rešenje koje „pokriva" sve 32-bitne konstante moglo bi, recimo, da glasi:

```
MOVX   R2, nižih_13_bita
SLR    R2,13
MOVX   R1, viših_19_bita
ADD    R1,R2
```

Međutim, pošto se prevođenje programa (teorijski!) obavlja samo jednom a izvršavanje puno puta, ima smisla da se kompajler malo pomuči ali da izvršenje programa bude brže. U tom cilju on će ispitati konstantu koju treba upisati u registar i prema njenom 12. bitu generisati odgovarajući kod. Ako je dvanaesti bit nula, generiše se



Slika 1



Slika 2

```
MOVX   R1, viših_19_bita
OR     R1, nižih_13_bita
```

Ako je dvanaesti bit jedinica, prevod će biti

```
MOVX   R1, prvi_komplement_viših_19_bita
XOR    R1, nižih_13_bita
```

Dakle, u R1 je upisana komplementirana vrednost gornjih 19 bita, tj. sve jedinice su zamenjene nulama a sve nule jedinicama. Procesor će jedinicu na dvanaestom mestu protumačiti kao znak i proširiti ga na gornjih 19 bita. Po izvršenju XOR-a sve dolazi na svoje mesto, tj. gornji deo registra se ponovo invertuje i dolazi u originalno stanje. Pošto su donjih 13 bita registra bili nula, tu se jednostavno prepisuje ono što je potrebno. Sve to ilustruje slika 2.

86: Potrebno je ______ poteza
Ime i prezime ______
Adresa ______
Mesto ______
Kompjuter ______ Vreme ______
Identifikacioni broj: |_| |_| |_| |_| |_| |_| |_|
iz broja tel. god. rođ. po izboru

KAKO DOSTAVITI REŠENJE

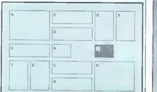
Rešenja *Pitalice* zajedno sa **obrazloženim** programima koje ste napisali pošaljite na uobičajenu adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd, tako da pristignu **pre** 26. jula 1993. Sve pisma sa korektnim rešenjima konkurišu za novčane nagrade u visini vrednosti 10, 7 odnosno 5 primeraka „Računara", dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača *Pitalice*. Identifikacioni broj dobijate tako što u prve tri kućice upišete **poslednje** tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru.
Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj. Osim običnom, rešenje ove *Pitalice* možete da pošaljete i elektronskom poštom posredstvom Sezama – konsultujte menije REDAKCIJA, PITALICE, REŠENJE ili iz komandnog moda pošaljite ličnu poruku korisniku pitalice (*mail write pitalice*).

ARGH!

Evo nešto težeg ali verujemo i zabavnijeg probleme. Na slici 3 prikazena je svojevrsna slagalice koja se sastoji od trinaest kvadrata i pravougaonika i praznog prostora veličine dva najmanja kvadrata. Figure su obeležena slovime A-M a na sličan način se obeležavaju i potezi. Iz polazne pozicija se na primer potezom I dolazi u poziciju sa druge slike. Ukoliko na taj način potez nije jednoznačno određen (na pomenutoj slici bi se na primer I mogao pomenti i udesno i ulevo), pored njega se piše malo slovo *l, d, g* ili *n*, što respektivno označava pomeranje ulavo, udesno, nagore i nadole. Na primer Id označava pomeranje kvadreta I udesno u prazan prostor, čime bi došao ispod pravougaonike F. Napomenimo da se pomeranje kvadrata I do ta pozicije broji kao dva poteza i obeležava sa I Id.

Cilj je pomeriti kvadrat A iz polazne pozicije u položaj ne trećoj slici – pri tome finalni položaj ostalih elemenata slagalice nije bitan, ali broj odigranih poteze predstavlja kriterijum vrednovanja rešenje: što menje poteze, to je rešenja bolje! Samo se po sebi razume da se elementi tokom partije ne smeju sklanjeli sa table, pomerati izvan nje i tome slično. Za eksperimantisanje će poslužiti kartonski model ili, ako imete PC računar, program







ARGH koji možete da preuzmete sa Sezama iz direktorijuma IBMPC\FUN. Ime asocira na uzvik koji ćete verovatno, ispuštati rešavajući problem ... ukoliko se na odlučite za kompjutersko rešavanje koje, videćete, daje gotovo spektakularne razultate.

Prednost izloženog rešenja u odnosu na osnovno je očigledna: ne angažuje se pomoćni registar, a broj instrukcija je prepolovljen. Verujemo da kompajleri zaista generišu takav ili sličan kod pri punjenju registra 32-bitnim konstantama. Postavlja se ne ravno pitanje zašto dizajneri mikroprocesore nisu predvideli tako elementarnu stvar kao što je punjenje registra konstantom. Radi se o prostoj statistici – savremeni RISC proceson se uglavnom projektuju za specijalizovane namane uz prethodnu detaljnu analizu postojećih programa koji se u datoj oblasti izvršavaju. I bez previše statistike se, međutim, može reći da se „veliki" brojevi u programima razmerno retko javljaju – kada ste zadnji put napisali nešto kao **for (i=986432586; i<992178542; i++)**? Sa druge strane „male" konstante su razmerno česte – **for (i=0; i<32; i++)** će se naći u mnogim programima! Ukoliko, dakle, svaku instrukciju zajedno sa implicitno zadatom konstantom svedemo na jednu reč, uštedećemo mnogo na pripremi naredbi i time višestruko ubrzati izvršavanja programa, po cenu da povremeno, kada zatreba neka veća konstanta, „potrošimo" dve instrukcije da je dovedemo u registar.

Činjenica da je opisano rešenje u dve instrukcije prilično teško za praćenje ide u prilog onima koji kažu da je priča o RISC procesorima zapravo priča o optimizacionim kompajlerima. Programi za RISC procesore se nar me retko (da ne kažemo nikad) pišu u asembleru – uz procesor se „pravi" i dobar C kompajler koji ubuduće prevodi i optimizuje sve potrebne programe. Čak se i taj kompajler ne piše na asembleru (kao što misle mnogi koji se sećaju prvih dana računarstva) već u C-u, pa se na neki način prevodi na drugom računaru i na kraju kompajlira samim sobom... a svi su se onoliko smejali kada je baron Minhauzen samoga sebe podigao vukući se za kosu! Optimizacije nalik na opisanu su tek manji deo „umetnosti" RISC C kompajlera – glavna stvar je preklapanje instrukcija i njihovih ciklusa, tj. mešanje instrukcija tako da rezultati budu isti ali da se izgubi što manje ciklusa. Na kraju se dobijaju potpuno nerazumljivi mašinski programi koji, na čuđenje svakoga ko ih pogleda, rade korektno, ali mnogo mnogo brže od bilo čega što bi „živ čovek" mogao direktno da isprogramira!

Prvu novčanu nagradu u vrednosti 10 primeraka „Računara" (više nećemo navoditi iznose jer ih inflacija između pisanja teksta i izlaska broja čini besmislenima) dobio je **Stojan Miloradović** iz Beograda, autor po našoj oceni najkompletnijeg rešenja ovog problema. Preostale dve nagrade smo izvukli – drugu novčanu nagradu u vrednosti 7 primeraka „Računara" dobio je **Nenad Orlović** iz Novog Sada, a treću (vrednost 5 primeraka „Računara") **Đorđe Lavadinović** iz Beograda. Čestitamo i ostalim rešavačima koji su uspeli da problem reše primenom samo dve instrukcije: *Nenad Čutunć, Jan Filčik, Miloš Prvulović* i *Janko Stamenović*.

Ovaj zanimljivi problem predložila je *Jelica Protić*.

677 [ALGO] Za ovu godinu smo malo zakasnili ali biće još godina – na slici 1 dat je program koji za zadatu godinu određuje datum na koji pada Pravoslavni Vaskrs. Datum se ispisuje po gregorijanskom kalendaru [Preuzeto sa Sezama prilog **Dragiša Kramanović** (*dragisak*)]

678 [OS MSDOS, PC] Nova verzija DOS-a uvek daje puno materijala za ovu rubriku. Primećeno je, na primer, da posle instalacije MS DOS-a 6.0 datoteka COMMAND.COM u *root* direktorijumu dobija *read only* atribut. Ovo smatramo korisnom inovacijom pošto se više puta dešavalo da neki početnik otkuca DEL *.* i obriše COMMAND.COM posle čega se sistem ne može podići bez DOS diskete. Ako ta disketa nije pri ruci (a početnici je i pored svih saveta u priručnicima često nemaju) ostaje samo da se „krivac" pita zašto je benzin tako skup.

Pomenimo i jednu anomaliju – u *boot* sektoru disketa formatirane sa FORMAT A: /U piše MSDOS5.0 umesto MSDOS6.0. Neko u Microsoft-u je očito zaboravio da podesi odgovarajući string [Preuzeto sa Sezama prilog **Dragan Jovičević** (*djovicevic*)]

679 [OS MSDOS, PC] Jedan od važnih noviteta MS DOS-a 6.0 je SENTRY opcija komande UNDELETE – po njenom aktiviranju DEL će datoteke, umesto da ih briše, kopirati u poseban direktorijum i tamo držati neko vreme dok prostor na disku ne počne da se smanjuje. Ukoliko se u međuvremenu predomislite, datoteka može da se vrati u originalno stanje bez ikakve mogućnosti greške koja inače „promakne" svakom UNDELETE programu. Skoro kao na *Novell*-u. Budite međutim oprezni dok pokazujete ovu opciju prijateljima: lako je moguće da trik ne uspe. Ukoliko naime arhiv atribut datoteke nije setovan (tj. ako datoteka nije menjana posle zadnjeg *backup*-a) SENTRY je **neće** sačuvati. Da biste poništili ovu konvenciju, treba da ugradite red **archive=FALSE** u datoteku UNDELETE.INI [Preuzeto sa Sezama prilog **Zoran Milosavljević** (*zormi*)]

680 [OS MSDOS, PC] Još malo nedokumentovanih komandi DOS-a 6.0. Sa **InstallHigh=Prog.Exe** program se instalira u visoku memoriju direktno iz CONFIG.SYS-a prema uputstvu za ovakvo pokretanje je zadužena komanda INSTALL, ali ona šalje program u konvencionalnu memoriju.

Druga zanimljivost je komanda TRUENAME (postojala je, doduše, i u prethodnoj verziji DOS-a) koja, reklo bi se, transformiše ime datoteke u najkraći mogući apsolutni oblik. Recimo komanda

TRUENAME C:\DOS\..\UTIL\TEST\..\..\TEKST\RAZNO\.

daje **C:\TEKST**

Pregledom datoteke IO.SYS nailazi se i na komandu MULTITRACK nepoznatog značenja. U CONFIG.SYS u možete da napišete MULTITRACK=ON i neće biti prijavljena nikakva greška; nismo međutim primetili da se išta promenilo. Nije jasna ni uloga komandi COMMENT (možda sinonim za REM?) i PROTMAN [Preuzeto sa Sezama prilog **Bojan Pavlović** (*pavbok*)]

681 [OS MSDOS, PC] Jeste da se DOS 6.0 tek pojavio, ali se već priča i o njegovom nasledniku. Tokom same promocije, Bil Gejts (*Bill Gates*) je najavio neke od planiranih specifikacija. Ovaj proizvod, čije je kodirano ime *Chicago*, trebalo bi da izađe za oko godinu dana i da se „zove" MS DOS 7.0. U pitanju će, po Gejtsu, biti „najveća promena u istoriji DOS-a". Novi operativni sistem koristiće sve prednosti 32-bitnog procesora uključujući i multitasking – korisnik će moći da pokrene više aplikacija u isto vreme. Odgovarajući na pitanja novinara o daljoj budućnosti DOS-a, Gejts je rekao da će DOS postojati još mnogo godina i da će ga *Microsoft* stalno unapređivati. Ipak predvideo je da će za pet godina prosečan PC računar raditi pod *Windows NT* „ispod" koga neće biti potreban DOS kao sada za *Windows 3.1*. *Microsoft* nema nameru da prisiljava korisnike da pređu na grafičko okruženje, ostaviće tržištu da o tome odlučuje.

682 [STAM, LJET4] Prilikom puštanja u pogon *LaserJet*-a 4 obratite pažnju na držač koji se nalazi unutar kućišta, blizu izlaznice za papir sa zadnje strane, kod napajanja. On tokom transporta pridržava sklop za fiksiranje stranica. Ako zaboravite da uklonite taj držač štampač će raditi ali će teksi biti ružniji nego što može da bude jer će se toner razlivati [Preuzeto sa BIX-a, autor *rsheater*]

683 [KOMS TDEBUG, PC] Ako verzija *Turbo Debugger*-a koju ste dobili uz *Borland C* pravi probleme sa *Windows*-om 3.1, poslužite se *Borland*-ovim *compatibility kit*-om. Zamenite WINDEBUG.DLL sa TDWIN.DLL i sve bi trebalo da radi potpuno normalno. Novije verzije *Turbo Debugger*-a (na primer ona koja se isporučuje uz *Borland Pascal 7.0*) savršeno „sarađuju" sa *Windows 3.1*.

Datoteku TDWIN.DLL *Borland* je stavio u javno vlasništvo i može se preuzeti sa Sezama [Prilog **Ivan Skart** (*skert*)]

684 [PJ C, Unix] Krupan bag C kompajlera na *SCO Open Desktop*-u. Neka je definisan string *pera* koji sadrži konstantu „Pera i Mika su legende", a zatim izvršite **sscanf (pera, „%s%[^\0]", mika, laza)**. Sadržaj *mika* će biti „Pera" što je očekivano, ali će *laza* ostati tamo gde je i bio. Funkcija vraća 1.

Isti primer preveden pomoću *Microsoft C*-a 6.0 radi besprekorno [Preuzeto sa Sezama prilog **Pavle Peković** (*ppekovic*)]

685 [BIOS, PC] Neki noviji AMI BIOS-i među ostalim opcijama imaju i (najčešće nedokumentovanu) stavku *Daylight savings* koja može da se uključi ili isključi. Odgovor na pitanje šta ovaj parametar znači potražili smo u okviru dokumentacije čipova MC146818 i MC146819 koji na većini savremenih PC računara „izigravaju" časovnik realnog vremena. Pokazuje se da je bit 0 njihovog registra 0Bh odgovoran za letnje vreme. Ako je bit setovan, poslednje nedelje u aprilu vreme će se sa 1:59:59 AM promeniti na 3:00:00 AM dok će poslednje nedelje u oktobru posle 1:59:59 AM nastupiti 1:00:00 AM. Ako je bit postavljen na nulu, ovi „vremenski skokovi" su isključeni.

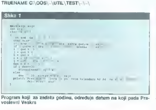
Obzirom da se kod nas na letnje/zimsko vreme **ne** prelazi poslednje nedelje u aprilu odnosno oktobru, odgovarajuću opciju treba ostaviti u podrazumevanom stanju OFF. Vreme ćete, kao i do sada, morati ručno da podešavate [Preuzeto sa Sezama prilog **Milan Stanić** (*mstanic*)]

686 [ASEM, PC] Jeste li poželeli da nešto iscrtate iz asemblerskog programa? Primer sa slike 2 može da bude taj prvi korak. [Preuzeto sa Sezama prilog **Ivan Milenković** (*mazi*)]

687 [SIS, PC] Pisanje iole ozbiljnijeg komercijalnog programa ne može se ni zamisliti bez obrade takozvane „kritične greške". Radi se o tome da će DOS, ako naiđe na nečitljiv sektor, „oseti" nezatvorenu klapnu disk drajva, neformatiranu disketu ili nešto slično, ispisati čuvenu poruku *Abort, Retry, Ignore* i to van ikakve kontrole tekućeg programa. Posledice se mogu zamisliti, naročito ako je računar bio u grafičkom modu. Zato je pogodno preuzeti obradu kritične greške i tretirati je kao i svaku drugu, na primer ispisujući sopstvenu poruku i čekajući da korisnik otkloni problem kako bi se operacija uspešno ponovila. Pri tome će vam pomoći program sa slike 3, jedno *bcc critical c handler.asm* bi trebalo da demonstrira kako sve to radi [Preuzeto sa Sezama prilog **Mirjan Jovanović** (*mjova*)]

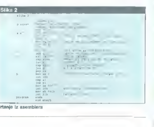
688 [TEHP, PC] Na tržištu je sve više IBM PS/2 računara pa komercijalni programi više ne mogu da dopuste „luksuz" da o njima ne brinu. Na slici **4** je program koji će detektovati *Micro Channel* računar. Prevedite ga *Microsoft*

Slika 1



Program koji za zadatu godinu određuje datum na koji pada Pravoslavni Vaskrs

Slika 2



Crtanje iz asemblera

Slika 3

```
[illegible]
```

Program za preuzimanje obrade kontrola kritične greške

C-om 7 0 konsteći komendu CL /W4 IME C [Preuzeto sa BIX-a autor (*bleszczak*)]

689 [PJ.CLIPPER, PC]. Često parametrima procedura treba dodeliti neka inicijalna vrednosti, za šta se obično koristi TYPE() i PCOUNT() na primer

Par3 – IIF (PCOUNT() >= 3, Par3, 'Default')

Slika 4

```
[illegible]
```

Program za detekciju *Micro Chennel* arhitekture

Postoji međutim još jedan zanimljiv način – recimo da je podrazumevene vrednost 1

FUNCTION Demo
PRIVATE p1,p2,p3,p4
STORE 1 TO p1,p2,p3,p4
PARAMETERS p1 p2 p3,p4

Metod je nekako primereniji starom *Clipper* u (*Summer '87*) gde se deklaracija PARAMETERS češće koristila [Preuzeto sa ALT BBS a prilog **David Jakalić**]

690 [PJ CLIPPER, PC] Većina ozbiljnih *Clipper* programa gradi komunikaciju sa korisnikom na centralizovanoj proceduri za skaniranje tastature Takva procedura se lako poziva iz *TBrowse*-a i GET podsistema *AChoice* i *Menu To* se mogu zameniti novim funkcijama ali je pravi problem *MemoEdit* Evo kako se on može rešiti

Na slici 5 je korisnička funkcija za *MemoEdit* u okviru koje se skanira tastatura uz prosleđivanje nekih od primljenih tastera „pravom" *MemoEdit*-u Pri tome se u bafer tastature „podmeće" i jedan kod koji *MemoEdit* sigurno neće razumeti pa će kada na njega naiđe vratiti kontrolu korisničkoj funkciji I tako u krug [Preuzeto sa Sezame prilog **Slobodan Kandić** (*kandic*)]

691 [PJ PARADOX_ENGINE, PC] *Paradox Engine* je sve popularniji među autorima programa koji operišu sa bazama podataka – mogućnost da programi pisani u paskalu ili C-u lako i efikasno operišu sa velikim bazama podataka za mnoge je sasvim dovoljna da ozbiljno porazmisle o preorijentaciji na ovaj paket Naravno čuda su u ovom svetu retka pa i *Paradox Engine* ima razne nedostatke Oduvek se smatralo da se jedna baza podataka ceni po tome koliko brzo indeksira pretražuje i sortira pa ćemo se u ovom prilogu baviti baš tim svojstvima *Paradox Engine*-a

Primarni indeks se formira na osnovu čistog uporeĎivanja stringa Pri tome se koristi čist ASCII redosled – izgleda da nema mogućnosti da se formira sopstvena tabela ključeva (što bi pomoglo pri YUSCII sortiranju) ni indeksiranja po vrednosti nekog izraza (što recimo *Clipper* omogućava)

Primarni indeks se formira po poljima sleva na desno što znači da se ako je potreban primarni indeks po trećem polju mora formirati i indeks po prva dva ili promeniti redosled polja Ne postoji mogućnost dupliranja ključeva (javlja se graška *key violation*) tako da se mora strogo paziti šta ulazi u indeksiranu tabelu To je ujedno i prva ozbiljna zamerka čitavom sistemu

Sekundarni indeks može da postoji tek ako postoji primarni što je takođe neprijatno jer često ne postoji logička potreba za primarnim indeksom u kome se ključevi ne smeju ponavljati već se on mora veštački formirati (uz utrošak prostora i vremena!) samo da bi *Paradox Engine* bio „zadovoljan" Postoje dve vrste sekundarnih indeksa SECONDARY i INCSECONDARY Prvi se ažuriraju samo prilikom otvaranja (kreiranja) datoteke a drugi se automatski održavaju dakle stalno su ažurni bez obzira na promenu podataka SECONDARY indeksi se retko koriste uglavnom ako se radi sa bazama iz kojih se podaci crpe umesto da se ažuriraju (recimo baza poštanskih brojeva) O INCSECONDARY indeksima nema mnogo šta da se kaže – uglavnom rade onako kako se od njih očekuje

Kompozitni indeksi se po načinu kreiranja razlikuju od sekundarnih po tome što se mapira novo polje koje je sastavljeno od više polja Pošto u bazi može da bude najviše 255 polja brojevi 1 – 255 su rezervisani za njihove redne brojeve (*field handle*) a brojevi preko 256 se koriste za mapirana polja Ova polja su logička slika nekoliko polja iz tog sloga jer se pod jednim rednim brojem dobija slika više polja To je samo jedan korak bliže indeksima po izrazu koje da ponovimo *Paradox Engine* ne podržava

Slika 5

```
[illegible]
```

Korisnička funkcija za *MemoEdit* u okviru koje se skanira tastatura

Nastavlja se na strani 77

Samo indeksiranje obavlja se izuzetno brzo, što je na neki način i logično kada se uzmu u obzir izložena ograničenja. Primarni indeksi su jako kratki – primarni indeks jedne datoteke od skoro 4 megabajta imao je samo oko 55 kilobajta. Na žalost, sekundarni indeksi koji su u praksi neophodni znatno su duži, tako da taj indeks, zavisno od tipa polja, može da bude dugačak koliko i sama baza podataka ili čak duži od nje. Pouzdanost čitavog sistema je sasvim zadovoljavajuća.

Na raspolaganju su brojne funkcije za rad sa bazom podataka i njenim indeksima, ali njihova upotreba nije naročito zgodna. Svaka funkcija, naime, vraća kod greške koja je možda nastupila i taj broj se mora odmah testirati, što uništava kako izgled tako i preglednost koda. Pozivi nekih funkcija su znatno različiti od onoga što je uobičajeno za C – na primer umesto **strcpy(char *dest, char *source)** koristi se **int PXRecBufCopy (RECORDHANDLE fromHandle, RECORDHANDLE toHandle)**. Ove nedoslednosti verovatno potiču od toga što se iste funkcije koriste i za vezu sa paskalom.

Sve u svemu, ima se utisak da je *Paradox Engine* „istrgnut" iz nekog većeg paketa bez dovoljno vođenja računa o posledicama. Kao da se članovi *Borland*-ovog *data base* tima nisu dovoljno udaljili od programiranja na bejziku! (Preuzeto sa Sezama, prilog **Miljan Jovanović** (*mjova*))

692 [COMM, VAX] U tekstu „Pošta pod sanckijama" iz „Računara 90" rekli smo da je korisniku koji sa VAX-om komunicira posredstvom modema najjednostavnije da poruku pripremi kod kuće, pozove VAX, otkuca MAIL SEND, ime primaoca i temu, a zatim izvrši *ASCII upload* poruke. Postoji međutim i pogodniji način koji veoma podseća na Sezamov *exec*. U okviru VAX-ovog LOGIN.COM-a treba definisati

$ exec:==run /input=exec.com /output=exec.log sys$system loginout.exe

Na PC-ju treba pripremiti ASCII datoteku sa komandama i tekstovima poruka koja veoma liči na ono što bismo kucali interaktivno, samo na onim mestima gde u interaktivnom radu počinjemo unos u editor u datoteci pišemo $DECK a na mestima gde bismo napustili editor kucajući *Ctrl Z* stavljamo $EOD (to su inače komande koje je bazično koristio čitač papirnih kartica). Mogući izgled tako dobijene datoteke EXEC.COM je

$mail	**$EOD**
send	**send**
Adresa1	**Adresa2**
Tema1	**Tema2**
$DECK	**$DECK**
Tekst prve poruke	**Tekst druge poruke**
itd	**itd**
	$EOD
	exit

Ostaje još da ovako pripremljenu datoteku pošaljemo koristeći DSZ/GSZ na PC-ju i RZ-A na VAX-u i da otkucamo EXEC. Poruke se šalju u pozadini, a mi za to vreme možemo normalno raditi. Na kraju treba proveriti datoteku EXEC.LOG u kojoj će se naći podaci o eventualnim greškama. (Prilog **Ljubomir Josifovski**)

693 [TERM, COMM] Vlasnici modema dobro znaju da je :-) osmeh, :-(nezadovoljstvo, ;-) ironija i tome slično. Međutim, ovih simbola ima mnogo više, toliko da se o njima može napisati knjiga. To je uradila firma *Peachpit Press, 2414 Sixth St. Berkeley, CA 94710*, telefon 510 548-4393 – knjiga se zove „*The Smiley Dictionary*" i košta 7 dolara. Lepa uleznica u svet računarskih komunikacija i BBS-ova!

I TO SE DOGAĐA

KVARLJIVI ČIPOVI

Iskustvo inženjera Bojana Mihajlovića iz Smedereva dokazuje staru istinu da u elektronici ništa nije nemoguće i da se upornost uvek isplati. Evo njegove priče o tome kako je otkrio sistemsku grešku u jednoj seriji IBM matičnih ploča i kako je – rešio problem

Ovim pismom bih želeo da preko vašeg časopisa upoznam širu javnost o onome što smo kod nas u SARTID-u uspeli da postignemo.

Po struci sam elektrotehničar za računare i radim na poslovima održavanja računara u preduzeću „SARTID 1913" (Železara u Smederevu – nap. red).

U proleće 1989. godine tadašnji MKS je kupio od firme INTERTRADE, tada ovlašćena za prodaju i servisiranje IBM-ovih proizvoda, 15 konfiguracija personalnih računara tipa PS/2 IT, model 50-061.

Početkom 1992. godine ovi računari su počeli, tako reći „serijski", da „crkavaju", što me je navelo da počnem da pronađem razlog za to.

Poznato je da je kod svih PC računara najbitniji element u konfiguraciji matična ploča. Ona objedinjuje kako fizički tako i logički pojedine komponente (bazične i periferne) raznorazne kartice (video, mrežna, modemske, faks kartice), periferijske uređaje (kao što su hard-disk, flopi-disk jedinice, tastatura), operativnu memoriju. Zbog svoje važnosti matična ploča mora da bude najsigurniji element u konfiguraciji PC računara. Stoga se svi proizvođači trude da u njih ugrade što pouzdanije komponente koje treba da obezbede što duži vek korišćenja računara.

Odavde pa nadalje će biti reči o matičnoj ploči IBM-ovog PS/2 IT računara, model 50-061, sa procesorom 80286. Ovu matičnu ploču je potpisao direktno IBM („COPYRIGHT IBM CORP 1987"). Ovakve matične ploče, kao i kompletan hardver PC računara, zbog visokog stepena tehnologije izrade moguće je popravljati i uspešno servisirati samo u namenskim IBM-ovim servisima (možda je čak bolje reći u laboratorijama) koji imaju visoku tehnologiju rada (savremena sredstva za rad, alata i uređaja za dijagnostiku neispravnosti).

Da bi se na ovim matičnim pločama izvodili bilo kakvi zahvati, potrebno je izuzetno poznavanje kompletnog PC računara, kako hardvera tako i softvera sa celokupnim operativnim sistemom.

Prilikom nabavke ovih PC računara uz njih nisu dobijene nikakve tehničke šeme niti bilo kakva tehnička uputstva za održavanje i popravke hardvera računara. Takođe ne postoji ni odgovarajući TEST uređaj za dijagnostiku koji bi olakšao pronalaženje neispravnosti. Postojale su jedino softverske test procedure za testiranje pojedinih delova računara.

Kada je u julu 1992. godine u naš servis stigao prvi neispravni računar ove serije, to je bilo naše prvo upoznavanje sa ovim tipom računara. Stigao je sa greškom koja se manifestovala na sledeći način: računar po uključenju podigne sistem i sve funkcije normalno odrade, ali posle određenog vremena blokira (ode u HALT stanje), tj. sistem se ukoči i dalji rad je onemogućen. Vremenski intervali posle kojih dolazi do blokiranja nisu uvek pravilni, tj. idu od 30 sekundi pa sve do nekoliko sati.

Upravo kod ovakvih kvarova je najteže otkriti i locirati neispravnost. I pored testiranja i pokušaja da se neispravnost izazove namerno, tj. pokušaja da se „zaguši" neki od perifernih uređaja (npr. non stop pristupanja hard-disku, flopi disk jedinici ili unošenja podataka sa tastature), čak ni namenske softverske test procedure za testiranje pojedinih delova računara nisu pomogle da se „uhvati" bilo kakva logika blokiranja računara (javljale su da su svi podsistemi ispravni).

Manifestacija kvara nas je navela da je neispravnost ipak na najslabijem delu računara – na matičnoj ploči. Tek tada smo se našli na početku problema.

Da bi se zaštitio od neovlašćenog kopiranja, IBM koristi neke svoje interne oznake za označavanje VLSI i drugih kola na svojim matičnim pločama. Taj njihov interni kodovnik nije poznat široj javnosti, što je još više otežalo pronalaženje neispravnosti. Bili smo oslonjeni samo na neka ranija iskustva stečena u radu sa drugim tipovima PC računara i na sopstvena znanja. Nakon višemesečnog rada na ovom problemu, posle mnogih testiranja, raznoraznih merenja, upoređivanja i praćenja signala između pojedinih elemenata na matičnoj ploči, februara meseca 1993. godine najzad je došlo do pozitivnih rezultata. Konačno smo mogli sa sigurnošću da pokažemo na određeni element i kažemo: „TO JE NEISPRAVNO". Ustanovljeno je da se neispravnost celokupne matične ploče krije u nekorektnom radu jednog integrisanog čipa.

U međuvremenu su računari ovog modela (koji u sebi imaju ugrađenu istu ovakvu matičnu ploču) počeli masovno da otkazuju – do danas je prestalo da funkcioniše ukupno 9 računara. Mnogi računari („model 50-061") su nošeni po raznoraznim servisima po Beogradu, ali ni jedan nije popravljen, već su svi nudili prodaju i ugradnju novih matičnih ploča po ogromnim cenama.

Prema oznakama na neispravnom čipu nismo uspeli da utvrdimo o čemu se zapravo radi, a ni u jednom katalogu koji mi posedujemo njega nije bilo. Konsultovane su mnoge privatne firme koje nabavljaju komponente, ali niko nije mogao da nam pomogne. Posle dublje analize funkcionisanja ovog elementa, njegove uloge u radu same matične ploče te konkretnih merenja signala na svim njegovim nožicama (ulazima i izlazima) rešili smo da napravimo emulaciju ovog čipa, tj. da emuliramo njegovo funkcionisanje – bolje rečeno, da dovedemo odgovarajuće signale na odgovarajuća mesta na matičnoj ploči i tako postignemo njeno pravilno funkcionisanje.

To nam je pošlo za rukom. Nestandardni čip, do kojeg je teško doći na domaćem tržištu, uspešno je zamenjen sa dva elementa do kojih je mnogo lakše doći i koji su uz sve to dosta jeftiniji. Sada je umesto jednog trebalo ugraditi dva elementa. U tu svrhu projektovana je pločica na koju će se montirati (lemljenjem) ova dva elementa, a sve to skupa treba da se ugradi na mesto neispravnog čipa. Prilikom projektovanja pločice vodilo se računa da ona zajedno sa svojim elementima predstavlja funkcionalnu i skladnu celinu, tj. da se pločica svojim dimenzijama (gabaritom) i načinom ugradnje prilagođava ostalim komponentama na matičnoj ploči.

Pločica je dimenzija 25x25 mm. Postavljena je pod uglom od 90 stepeni u odnosu na matičnu pločnu. Ovim je izbegnuto pregrevanje pločice i elemenata na njoj, jer se pokazalo da čipovi visoke integracije (VLSI) koji nisu pravljeni u CMOS tehnologiji troše dosta struje, samim tim oslobađaju dosta toplote, a nalaze se u okolini pločice. Pločica je montirane na matičnu ploču pomoću odgovarajućih nožica. Ovakva montaža obezbeđuje potrebnu stabilnost i funkcionalnost.

Pravilno funkcionisanje ovog elementa je preduslov da matična ploča ispravno radi, a sastoji se u sledećem: prilikom samog startovanja računara na ulaz ovog elementa stiža određen signal koji je postavljen u prethodnom stepenu, a na izlazima dobijamo konstantne signale koji na tom logičnom nivou ostaju za svo vreme dok računar radi (ne menjaju vrednost). Izlazni signali ovog elementa služe da postave početne parametre za rad susednog VLSI kola. Bez ovih signala pokretanje i rad samog PC računara je nemoguć.

Trenutno je na ovaj način uspešno osposobljeno 9 matičnih ploča. Od toga je 6 već ugrađeno u konfiguracije i besprekorno rade više od tri meseca.

Na kraju bih zamolio sve one koji su imali slične probleme sa ovim modelom PC računara, ili ih još uvek imaju, da mi se jave kako bih stekao uvid u učestalost ovakvih neispravnosti, eventualni drugačiji način manifestacije ove iste greške. Telefon: **(026) 732-348**

Unapred se zahvaljujem na saradnji.

Bojan Mihajlović
Smederevo

Tokom marta Sezam je „opslužio" 28 685 poziva, komunicirao sa korisnicima 7 146 časova, tj. gotovo 298 dana (s obzirom da mart ima 31 dan, u proseku je 9,6 nodova bilo *neprekid*no zauzeto), od čega je u *chat*-u provedano 1207 sati ili 150 radnih dana

SEZAM 22

SISTEM ZA MODEMSKE KOMUNIKACIJE

Priprema: Dejan Ristanović u saradnji sa moderatorima konferencija

Moderatori: Dejan Ba... (CIVILIZACIJA), Nikola Bošković (PC USER), Jovan Bulajić (FILES), Aleksandar Damnjanović (PC OS), D... (PC HARD), Slobodan Kaezić (ORKA), Vladimir Kostić (mnuje), Željko Krstić (VCEVI), Vladimir Ristanović (...), Ranko J...ović (RAČUNARI), Dalibor Lanik (ATARI ST), Bojan Pavković (PC UTIL), Pavle Peković (PC PROG), Dejan Ristanović (NOVOSTI, KOMUNIKACIJE), Bojan Tepavčević (FORUM) i Zoran Životić (SEZAM)

Korisnici su sa Sezama *download*-ovali 1,8 gigabajt podataka, dok je za isto vreme sistemu poslat 61 megabajt. Najviše poziva uputili su Igor Stanojević (*stigor*), Baša Stipčević (*basa*) i Miloš Đermanović (*eagle*), najviše vremena na vezi su proveli Vesna Denčić (*veca*), Željko Krstić (*zkrstic*), Irena Durmišević (*pahuljica*) i Srđan Jočić (*sjocic*), datoteka su najviše preneli Sanja Prus (*sprus*), Stanislav Dikić (*dikla*), Dragan Radaković (*daolid*), dok su najviše konferencijskih poruka poslali Dragan Nedeljković (*indragan*), Boris Malović (*mbole*) i Nenad Milenković (*wizard*).

Što se „kajnih vrednosti" tiče, tokom marta su u javne konferencije pristigle 9 234 poruke (skoro 9 megabajta teksta), gotovo sto toliko megabajta datoteka priključenih uz njih, 7 627 privatnih poruka (2,9 megabajta privatne pošte) i 5 255 poruka u grupama (1,7 megabajta teksta, ne računajući datoteke priključene uz te poruke).

NOVOSTI — Dejan Ristanović

U konferenciju **NOVOSTI** stizale su vesti sa BIX-a i nekoliko drugih stranih mreža, a počelo je i prenošenje jedne diskusione grupe sa USENET-a koja se bavi c++-om – ako interesovanja bude na nivou, mogu se prenositi i druge slične diskusije. Naši korisnici zbog sankcija ne mogu slati poruke u ove internacionalne konferencije, ali i samo praćenje diskusije i aktuelnih problema svetskih programera nije za bacanje. U temu *world.news* i dalje pristižu nekompjuterske vesti sa *ticker*-a poznatih stranih agencija – dobar deo tih vesti bavi se upravo situacijom u (bivšoj i sadašnjoj) Jugoslaviji. Ove vesti osim Aleksandra Bakića (*bale*) počeo je da prenosi na Sezam i Srđan Kusovac (*skusovac*).

SEZAM — Zoran Životić

Glavna tema konferencije **SEZAM** svakako je bilo proširenje kapaciteta sistema – početkom marta montirano je pet novih linija i promenjen broj telefona Sezama. Ovim proširenjem Sezam je „uhvatio priključak" sa najvećim evropskim BBS-ovima – najveći sličan sistem u Evropi, švajcarski „HITLINE Communications Ueli Maurer", ima 21 liniju. Što je mnogo važnije, naši korisnici sada mnogo lakše dobijaju Sezam – preko dana obično iz prve, a u tradicionalno najopterećenija ponoćna vremena posle desetak minuta zvanja. Nove linije i veći broj poziva uvek donose nov kvalitet diskusijama, što se potvrdilo već prvih dana po proširenju. Sezam je i dalje zasnovan na lokalnoj mreži PC računara sa dva servera: jedan od njih je *Compaq DeskPro 386* sa diskom od 330 megabajta, a drugi „klasičan" 386/33/8 MB RAM sa diskom od jednog gigabajta. Stanice su 15 AT i 386SX računara opremljenih *Ethernet* kompatibilnim mrežnim karticama i modemima *ViVa 2400 MNP* i *Discovery 2400 MNP*. Osim petnaest javnih linija, na raspolaganju je i jedna linija za održavanje sistema, kao i nekoliko lokalnih nodova u Redakciji Računara i Novinskom sektoru BIGZ-a; operativni sistemi su *Novell Netware 3.11* i MS DOS 5.0. U konferenciji Sezam je, osim stavlja zbog novih linija, bilo dosta diskusije o budućnosti Sezama, tj. jednoj stavci „srednjoročnog plana" koju još treba ostvariti – vezi sa drugim većim sistemima u zemlji i, po ukidanju sankcija, svetom.

KOMUNIKACIJE — Dejan Ristanović

Najaktivnije teme konferencije **KOMUNIKACIJE** su, kao i mnogo puta ranije, bile *modemi* i *kom.program*. Bilo je reči o tome da li faks modemi mogu da uključe računar u trenutku kada neko šalje faks, ako već najvećim delom ne mogu, zašto uopšte trošiti pare na njih. Pokazalo se da je glavna prednost i dalje slanje faksa, pošto se izbegava ispisivanje poruka na papiru i njeno skeniranje uređajem sumnjivog kvaliteta, ugrađenim u komercijalna faks uređaje – uz odgovarajući modem sav posao obavi tekst procesor. Zanimljivo pitanje bili su fosil drajveri koji se izvršavaju ispod nekih komunikacionih i BBS paketa, ali mogu da predstavljaju i faktor nestabilnosti čitavog sistema ... sam Sezam je u prvim danima svog rada komunicirao sa modemom posredstvom ... a ... se kasnije pokazalo neophodnim da se ... odgovarajućim specijalizovanim pol programima. Opcija &G (*Guard Tone*) je, ... neopravdano optužena za otežano uspostavljanje veze. Ukratko je opisan *Videotex* i CCITT V.23 protokol, razmenjena neka iskustva sa programom *TeleReplica*, predstavljen *patch* za *Procomm for Windows 1.0*, a zatim i *Procomm Plus for Windows 1.01*. *WinFax* je upoređen sa *BitFax*-om, savetovano da se koristi *WinFax Pro* a ne *WinFax Lite* (*Lite* je namerno oslabljena verzija koja se deli uz razne modeme), opisan UUENCODE/DECODE 5.21 format. U okviru teme *email.vax* diskutovali smo o X.400 pošti u Beogradu (Intel je u saradnji sa IRIS-om pustio u probni rad X.400 čvor u PTT-u) i raznim protokolima za razmenu podataka preko paketne mreže, kao i o dokumentaciji u kojoj se mogu naći opisi tih protokola. Tema *stran.bbs* bavila se zvanjem UBBG a iz sveta, pristupom *CompuServe*-u u svojstvu gosta, karakteristikama EXEC PC BBS-a i pretplatom na njega, predstavljen je ažuran spisak ... PD SW programa sa stranih sistema i razmenjeno više poruka koje se bave bezbednošću elektronske pošte, na primer primenom programa PGP čija je nova verzija (2.2) takođe na Sezamu. U temi *yu.bbs* bilo je dosta poruka, dosta polemike ... bavili smo se raznim događajima na Durlan, Hobbiton, Pingvin (naročito navodno tvoračnom konferencijom erotika), Politika, Rhinoceros i drugim BBS-ovima. Razgovarali smo i o BBS programima, odnosno o odluci koju svaki novi *SysOp* najteže donosi – da li se opredeliti za *PC Board*, *QBBS*, *Remote Access*, *UBBS*, *Ultra BBS*, *Waffle*, *Wild Cat*... Standardno dobro posećena tema *rtv.ptt* se, osim uvođenjem tonskog biranja, bavila i pojmom REN (*Ringer Equivalence Number*) koji određuje broj telefona koji se mogu vezati paralelno na jednu liniju a da svi zvone kada neko zove, zatim satelitskim časovnicima koji se automatski doteruju prema signalu ulovljenom sa neba, elektronskim časovnicima koji stalno zure i kasne zbog toga što je naša elektroprivreda izgubila referentnu frekvenciju koja je nekada stizala iz Austrije preko ..., sada službenih da ekovoda ... Glavne poslepriključenjem modema na satelitski telefon projekat je na kraju uspeo ... jedan novinar nam se svakodnevno javljao iz Italije preko satelitskog telefona ... kao pozivnim brojevima za razne države – ako vam neki zatreba a nemate imenik pri ruci, pogledajte HELP za komandu COUNTRY MS DOS-a 6.0: to su baš pozivni brojevi! U okviru teme *terminologija* osim „večitog" pitanja razlike između bauda i bita u sekundi diskutovalo se i o BBS simbolima – uz razne uobičajene „smajlije" pojavio se i nekoliko kreacija sa Sezama, npr. ( )|( )|( ) označava prevrtanje preko glave od sreće!

SPECIJALNE USLUGE

PODSISTEM TANJUG

Podsistem Tanjug omogućava ... Servis ... agencije TANJUG ... ekonomskim, društvenim, naučnim, sportskim ... iz zemlje i sveta. Servis neprekidno ... stalno ... registrovani korisnici ... mogu čitati i prenositi na svoje računare ... Pristup ovom podsistemu ... (MENU TANJUG) ... komandnog moda naredbom NEW.

Pristup podsistemu ... uslugama ... plaća se posebno. Za detalje ... obratite se Novinskoj agenciji Tanjug ... **Milena Sekulić**, tel. 625-722.

PODSISTEM BETA

Podsistem „Beta" obezbeđuje praćenje domaćih časopisa u elektronskoj formi. Časopisi pristižu na Sezam pre ... trenutka izlaska na kioske ... prema pretplatama korisnika mogu pregledati, čitati i prenositi na sopstvene računare. Pristup ovom podsistemu ostvaruje se iz menija (MENU BETA) i komandnog moda (naredba BETA). Trenutno se na ovaj način distribuira nedeljnik Vreme ... pristup je u probnom periodu slobodan.

Pristup podsistemu „BetaPress" je posebna usluga Sezama i plaća se posebno. Za detalje oko pretplate ... obratite se Marketingu lista Vreme. Kontakt: **Vojislav Milošević**, tel. 646-070.

PC PROG — Pavle Peković

Da li matični broj, broj tekućeg računa i drugi brojevi sa kojima se svakodnevno srećemo imaju svoja kontrolna sume, tj. da li se može proveriti njihova ispravnost i kako izgledaju odgovarajući algoritmi – to je bila hit tema u martovskim danima konferencije **PC PROG**. Burno je bilo i u temi *jezici* gde smo čitali razloge za i protiv COBOL-a, priču o aktuelnim verzijama tog jezika za PC, kao i spekulacije o COBOL-u za *Windows*. Paskal programeri su pričali o grafičkim formatima, skeniranju, tastaturi, kontroli štampača, inverznom ispisu teksta, skrolovanju i kontroli grešaka. Zanimljivo je bilo čitati i razmenu iskustava dva srednjoškolska nastavnika o *Turbo Pascalu 2.0* na računaru TIM kojeg ima dosta u našim srednjim školama. *Paradox Engine* je ovog meseca privlačio pažnju ljubitelja C-a, a nije izostalo ni „prepucavanje" korisnika *Microsoft* i *Borland* C-a o kvalitetu kompajlera. Pričalo se o kontroli čipa 8254 i implementaciji logičkog XOR-a. Objektno orijentisani C ovci su čitali poruke sa USENET konferencije *comp.lang.c++*, zahvaljujući Aleksandru Bakiću (*bale*). *Fox* i *Clipper* programeri su konačno naučili da u miru žive zajedno, što je donelo dosta zanimljivih poruka o mogućnostima ova dva programska jezika. Pisalo se i o problemima sa *upgrade*-om *Clippera 5.01*, „nagovaranju" *Clipper*-a da umesto Y/N koristi D/N, iskustvima sa preprocesorom, problemima sa memorijom... Hit su bile i razne dodatne biblioteke za *Clipper*.

PC USER — Nikola Bošković

I ovog meseca je u konferenciji **PC USER** bilo veoma živo. U temi *editori* se diskutovalo u kojem editoru je najpogodnije razvijati sopstvene aplikacije. Takve rasprave kao obično izazivaju burne polemike o tome koji editor šta pruža, tako da se uvek iznese po neka nova stvar o aktuelnim editorima. Cela polemika je počela tako što je jedan korisnik pitao kako da izbegne „nazubljivanje" programa koji iz *Borland*-ovog rezidentnog HELP-a prenese u editor. U igri su bili *QEdit* i *Brief*; i jedan i drugi editor imaju mogućnost isključivanja te opcije, ali neki od korisnika misle da im je to prednost jer je program mnogo pregledniji kada je „nazubljen". Korisnici su davali predlog da se koristi *Borland*-ova integrisana okolina, dok su drugi smatrali da je mnogo povoljnije koristiti isti editor za sve jezike. *Brief* što se toga tiče pruža velike mogućnosti. Pojavom paketa TTF4DOS (*True Type Fonts For DOS*) tema *word.perfect* je živnula: pomenuti paket omogućava korišćenje *True Type* fontova kako u WP-u tako i u *Word*-u. Uz pomoć ovog paketa sve fontove koje imate za *Windows* možete koristiti u DOS verziji navedenih paketa, a utisci korisnika Sezama su veoma pozitivni. Što se tiče standardnih problema *WordPerfect*-u (operacije sa blokom kao što su *move*, *copy*...), oni se rešavaju veoma brzo, na opšte zadovoljstvo korisnika. U temi *cad* videli smo kako se običan ASCII tekst može ubaciti na određena mesta u crtežu, veliku pomoć pri tome pruža *Lisp*. Tema *razno* je bila, kao i obično, veoma živa. Diskutovalo se o programima za učenje slepog kucanja, kao i o pravilnom rasporedu prstiju na tastaturi za slepo kucanje. Uz burnu diskusiju među korisnicima dobili smo i nekoliko interesantnih tekstova. Jedan od njih se odnosi na vrste tastatura, tj. QWERTY (QWERTZ) standard i njegove modifikacije nastale kao rezultat višegodišnjeg ispitivanja reči koje se najčešće koriste. Tako se došlo do interesantnog rešenja da se u srednjem redu po trenutnom rasporedu slova može otkucati svega 50-ak reči, dok se po tom novom rasporedu može otkucati čak oko 3000 reči! Ta tastatura je dobila ime DSK (*Dvorak Simplified Keyboard*): u srednjem redu su svi samoglasnici i to kod leve ruke, dok se najčešće upotrebljavani suglasnici nalaze kod desne ruke. Ovim rasporedom slova se postiže povećanje brzine rada daktilografa i značajno smanjenje broja grešaka.

PC UTIL

Bojan Pavković

Još od izlaska PKZIP-a 2.04 tema arhiveri je „zvezda" konferencije **PC UTIL**. Korisnici su odlučili da vrlo detaljno uporede arhivere, tako da se ovog meseca raspravljalo o tome gde koji od njih pravi prvi manje datoteke. S obzirom da arhiveri ZIP i ARJ to rade na različite načine, opet se diskusija svela na to koji je arhiver po tom pitanju bolji, kao i da li je to jedan od bitnijih parametra izbora arhivera. Diskutovalo se i o načinu na koji se datoteke šifruju pri arhiviranju, odnosno da li se lozinka igde upisuje i da li ju je moguće pročitati (zaključak je da nije – šifra bi se morala „napadati" nekom kriptoanalitičkom metodom). Nastavljen je razgovor o tome koji od programa za arhiviranje nudi veću mogućnost oporavljanja oštećenih arhiva, kao i kakva su iskustva korisnika. Utvrđeno je da pri korišćenju smanjene verzije PKZIP-a (junior) može nastati neispravno sažimanje spakujuća datoteka. Krajem meseca se pojavio novi arhiver SQZ (made in Europe) i odmah je upoređen s dva najaktuelnija (ZIP, ARJ). Prijatno iznenađenje je bilo da arhiver SQZ bolje sabija podatke nego ZIP i ARJ, ali je zato značajno sporiji i od jednog i od drugog, a posebno od PKZIP-a.

Za razliku od marta 1992. koji je bio u znaku Mikelanđela i istoimenog virusa, ove godine je sve to prošlo relativno tiho, mada su korisnici izjavili da je i sada bilo podataka koji su zauvek nestali zahvaljujući njegovom razornom dejstvu; možda je olakšavajuća okolnost što je 6. mart pao u subotu pa su računari u firmama bili isključeni. Za pogoršanje situacije pobrinuo se, na žalost, čuveni autor antivirus programa *McAfee*, jer je tada aktuelna verzija programa CLEAN imala probleme pri odstranjivanju Mikelanđela s diska, činjenica da se to moglo popraviti ručnom ispravkom „očišćenog" diska je malo značila manje iskusnim korisnicima. Docnije se pojavila nova verzija programa CLEAN, ali korisnicima koji su ostali bez podataka sve to nije mnogo pomoglo. Krajem meseca stigla je i informacija o virusu koji, kada je aktivan, simulira učitavanje operativnog sistema s diskete, tako da korisnik ima utisak da je „podigao" sistem sa diskete na kojoj sigurno nema virusa, dok on i dalje ostaje aktivan u memoriji. Diskusija u temi *tools* bile su iniciane pojavom dva programa domaćih autora. Prvi je BFIND, program za traženje datoteka Bojana Petrovića (*pbojan*), a drugi je CDP, program za promenu tekućeg kataloga Stanislava Šokorca (*ssokorac*). Programi su izazvali vrlo povoljne reakcije korisnika, a bilo je i primedbi i predloga kako da se poboljšaju, pa se uskoro mogu očekivati nove verzije. Poređenje ovih programa sa aktuelnim programima iste ili slične namene pokazalo je da su oni podjednako dobri, a u nečemu i bolji, tako da se ne treba čuditi ako na nekom računaru umesto programa FF zateknete BFIND ili umesto programa NCD CDP. Diskutovalo se o programu PCL koji neverovatno efikasno štiti računar od neovlašćenih ruku, ali ga se teško otarasiti – izgleda da ga ni formatiranje diska ne uklanja. Činjenica je da autor o tome upozorava u dokumentaciji i da je to u stvari dobra osobina programa takve namene, ali ipak dobro razmislite pre nego što ga instalirate. Diskutovali smo i o tome treba li koristiti *stealth* opciju QEMM-a, kakvi se sve problemi mogu javiti ukoliko se pri kopiranju diskete kopira samo deo sa podacima (ovu opciju imaju svi poznatiji programi za kopiranje disketa COPYQM, DISKDUPE), kako zvanične ispravke u programu QEMM utiču na rad *Windows*-a, konstatovali smo da je u nas najrasprostranjenija verzija *Windows-a 3.1* u stvari beta verzija (datum datoteka 17-02 1992), rešavali probleme sa merama u jeziku HPGL.

ORKA

Slobodan Kalezić

S obzirom da se prethodna konferencija **ORKA** napunila, startovali smo novu, treću po redu. Nova konferencija ima jednu temu više od stare – *obrazovanje*. Diskusija u ovoj temi započela je veoma burno, pošlo se od nastave programiranja u školama, ubrzo došlo i od nastave fizičkog (da li je u redu terati đake da se u interesu zdravlja bave sportom ako to ne žele?) a i drugih predmeta. Videli smo zadatke sa republičkog takmičenja u programiranju, kao i stu najboljih. Jozef Kiš (*jkmica*) je poslao više informativnih poruka o programerskoj Olimpijadi srednjoškolaca koja će biti održana oktobru meseca u Argentini. Saznali smo kako se biraju zadaci za takmičenje, kako se formira komisija za pregled i pregledaju zadaci, program boravka i takmičenja, pozvana je i naše zemlja. Dobili smo nastavni program jedne američke srednje škole – za razliku od naših programa, učenik tamo može da izabere oko polovine predmeta. Izgleda da bi takav program bio veoma odbro prihvaćen u našim školama, sudeći bar po živoj diskusiji da li i koliko treba učiti o računarima, odnosno koliko ima smisla terati đake nekog prirodnog smera da čitaju ruske klasike. Preneta je i rasprava sa BIX-a na temu korišćenja džepnih računara na ispitima: student sa boljim računarom gubi mnogo manje vremena na izračunavanja, a može u njega da smesti i podsetnik. Na nekim ispitima je zabranjena upotreba svih džepnih računara, na nekim samo HP 48, na nekim opet samo neke klase. Postavlja se i pitanje zašto neko na ispitu ne bi smeo da koristi računar koji inače koristi svaki dan? Zaključak diskusije je ipak bio zajednički – treba zadati takve zadatke da korisnici (moćnijih) računara ne budu u prednosti u odnosu na ostale, tj. simulisati razmišljanje a ne pamćenje. U temi *hp48sx* dobili smo interesantan prilog – kako uz pomoć priručnog materijala praviti razne neuobičajene ili nestandardne konektore? Klub programera se iz hotela Slavija preselio u prostoriju MZ „Nada Purić", Koče Kapetana 26. Sastanci su svake subote u 20 časova.

> ### KAKO POSTATI ČLAN
>
> Postupak učlanjenja počinje [illegible] – podesite parametre komunikacije na **2400 8N1** (ako imate MNP modem, isključite MNP) i pozovite (011)648-[illegible]. Posle [illegible] Sezam će tražiti prijavu.
>
> **Username**
>
> Na vama je samo da otkucate **NEW** (novi korisnik), a Sezam će vas dalje voditi kroz proces prijavljivanja.
>
> **Izbor imena**
>
> Jedan od prvih podataka koje treba da date je puno ime i prezime, a potom i pseudonim pod kojim ćete koristiti usluge Sezama. Jednom izabran pseudonim ostaje vaše trajno ime na Sezamu (ne može se menjati). Ne znači da ga treba pažljivo izabrati – predlažemo vam da još pre poziva opredelite za pseudonim koji bi trebalo od podataka ne vaše ime i prezime, ukoliko nemate ideju. Sezam će vam predložiti pseudonim sastavljen od vašeg prezimena i prvog slova imena. Pseudonim može da ima između 4 i 18 znakova (preporučuje se 5-8 slova, pošto više znakova znači više kucanja, kako za vas tako i za one koji sa vama komuniciraju) i u njemu se mogu naći isključivo slova i znak tačka.
>
> **Izbor lozinke**
>
> Sledeći korak je izbor lozinke (*password*) kojom štitite vaš račun – za razliku od pseudonima koji je javna informacija, lozinku znate samo vi i Sezam. Zato je izaberite pažljivo, naročito izbegavajte da bude jednaka pseudonimu ili da se sastoji od vašeg imena, nadimka ili imena nekog člana porodice – takve lozinke se lako pogode, što donosi različite probleme, pre svega onome čija je lozinka „ukradena". Izabranu lozinku docnije možete da menjate.
>
> **Izbor radnih parametara**
>
> Ostaje još da izaberete radne parametre (arhiver, protokol i slično) – ne brinite ako još ne znate šta je šta, pošto će Sezam uvek predložiti uobičajene vrednosti.
>
> **Unos ličnih podataka**
>
> Prilikom unosa adrese, telefonskog broja budite veoma pažljivi, jer netačno unesena adresa onemogućava kontakt Uprave Sezama sa vama, a samim tim i vaše učlanjenje. Unošenjem adrese i ostalih podataka završena je prva faza vašeg prijavljivanja na Sezam.
>
> **Redovno...**
>
> Sledećeg radnog dana bićete vam postato [illegible] osoba [illegible] Sezam [illegible]. Ukoliko želite da [illegible] pozovete sledećeg dana [illegible] Sezam će potvrditi [illegible] podatke. Pošto dobijete [illegible] pretplaćuje e [illegible] na osnovu [illegible] odgovarajući iznos:
>
> | | |
> |---|---|
> | za 1 mesec | 8 bodova |
> | za 3 meseca | 15 bodova |
> | za 6 meseci | 25 bodova |
> | za 12 meseci | 40 bodova |
>
> Vrednost boda je vezana za kurs nemačke marke i menja se jedanput nedeljno, svakog ponedeljka. Na uplatnici koju dobijate od redakcije upisani su u redu spoj vašeg korisničkog imena i trenutno važeći iznosi za svaki period pretplate, kao i rok do kog treba izvršiti uplatu po navedenim cenama. Na primer 160000, 300000, 500000 i 800000 do 17.04.93. Ako uplaćujete ne svojim uplatnicama, vrednost boda, odnosno cenu pretplate možete saznati sa Sezama [illegible].
>
> Nakon izvršene uplate [illegible] potvrdu pošaljite pismom na adresu: Računari [illegible] Vojvode Mišića [illegible] Beograd, faksom na telefon (011)648-140 [illegible] (011)647-955. Uprava će vam na dan prijema potvrde odobriti pristup sistemu, a sledećeg dana će na vašu adresu biti poslato Uputstvo za upotrebu Sezama, atraktivno štampana knjižica (160 strana) pomoću koje ćete upoznati sve mogućnosti sistema.
>
> **... i ekspresno učlanjenje**
>
> Oni koji ma se žuri mogu postati korisnici Sezama za svega nekoliko sati. Javite se na opisan način a onda, ne čekajući pismo, izvršite uplatu u pošti navodeći podatke: **Pošiljalac**: vaše ime, prezime i pseudonim koji ste izabrali; **Primalac**: BIGZ – Računari; **Svrha uplate**: Pretplata na Sezam; **Žiro račun**: 60802-603-23264, **poziv na broj** 08 108 5. Potvrdu o uplati pošaljite faksom do 13 časova (radnim danom), već u popodnevnim časovima ste punopravni korisnik Sezama.
>
> **Dvostruko prijavljivanje**
>
> Ukoliko tokom predstavljanja imate problema (npr. veza se prekine) pozovite ponovo, prijavite se navedeći kompletne podatke. **Pažnja**: ako ste se prijavili dva ili više puta, kao što smo korisničko ime koje je napisano na uplatnici.
>
> **Koliko važi uplatnica**
>
> Cene navedene na uplatnici važe samo ali navedenog roka. Na vim pretplatnicama dajemo **za uplatu** rok od sedam dana od trenutka prve prijave, i starim pretplatnicima od prvog upozorenja za produženje pretplate. Smatramo da je sedam dana sasvim dovoljno da vaše uplatnice stignu do vas. Ukoliko ne izvršite uplatu u tom roku, pretplatu treba uplatiti po cenama koje važe u toj nedelji u kojoj se vrši uplata. To ne znači da se morate ponovo prijavljivati – dovoljno je da pozovete Sezam, predstavite se izabranim imenom i lozinkom, sistem će ispisati ažurne cene pretplata.
>
> **Razgledanje sistema**
>
> Ukoliko pre prijavljivanja želite da razgledate sistem, na pitanje **Username**: odgovorite sa DEMO. Naći ćete se u demonstracionim [illegible] u kome možete da pregledate kompletnu strukturu sistema koja je identična „pravoj". Vreme DEMO prijave nije ograničeno, a vreme za jednu seansu je 5 minuta.

IGRE

Branislav Mijatović

Krajem februara otvorena konferencija **IGRE** naišla je na izuzetan prijem kod korisnika Sezama, što se može zaključiti i po broju poruka pristiglih u nju. U startu se razvila vrlo živa diskusija o potrebi izdvajanja teme FRP (*Fantasy Role Play*) iz postojeće teme *avanture*: zagovornici su bili vrlo uporni i ubedljivi, pa smo im izašli u susret. Među korisnicima su izgleda veoma popularne borbene simulacije leta, koje se igraju preko modema, u odgovarajućoj temi organizovana su čak i takmičenja, tako da ćemo uskoro znati ko su najbolji piloti Sezama. Ipak, ubedljivo najveće reakcije izazvalo je podsećanje korisnika na neke od odredbi Ugovora o korišćenju Sezama, pre svega o zabrani razmene piratovanog softvera. No, čak i bez te polemike nema sumnje da će konferencija IGRE biti jedna od najposećenijih na Sezamu.

FORUM

Bojan Tepavčević

Mart je u **FORUM**-u protekao u znaku sporadičnih osvrta na aktuelnu vojno-političku situaciju, Vens-Ovenov plan i slične teme. Nešto više komentara zavredio je beg Jezdimira Vasiljevića iz zemlje, baćevi o avionu koji je dan pre toga prinudno sletao u Beograd, pri tome, kažu neki, odneo sve – sve što, kao i prognoze o sudbini ostalih privatnih banaka kod nas. Već prvog dana po gazda Jezdinom odlasku pojavio se i vic: Kakva je razlika između Jezde i Dafine? Mesec-dva – pokazalo se, na žalost, da nije u pitanju vic. Korisnik kojeg smo u februaru zapamtili po anketi o kvalitetu vida među Sezamovcima, tokom marta je sprovodio anketu „da li se švercujete u autobusima GSB-a?", na šta je odbio odsta potvrdnih odgovora, propraćenih ličnim iskustvima korisnika.

CIVILIZACIJA

Dejan Balinda

Mart je bio drugi uzastopni mesec kako je oboren rekord po broju poruka u konferenciji **CIVILIZACIJA**. Uz to ova konferencija je nadmašila ostale svojom dinamikom i raznovrsnošću tema o kojima se razgovaralo, teško je i pobrojati o čemu je sve bilo reči, s obzirom da je prestiglo preko 1.000 poruka. Ipak, vredi reći da je posebno zapažena bila žustra diskusija „Za i protiv prava na abortus", koja je malo koga ostavila indiferentnim. Ne treba se čuditi što je na Sezamu dominiralo liberalno stanovište, a razlike su se, pre svega, odnosile na stepen do koga se u ovoj slobodi može ići. Plodna je bila i rasprava u donekle zapostavljenoj temi *umetnost*. Pokazalo se da nije sasvim jasno šta je to kič, gde počinje a gde prestaje. Konstantno je bio već gotovo uobičajeni pregled repertoara beogradskih pozorišta za naredni mesec, a zapaženi su i osvrti na pojedine nagrade za filmsku umetnost u SAD i Francuskoj. Jezičke zavrzlame i ovog su meseca oduzimale dah svojom lucidnošću i pikanterijom. Ukratko, bio je to veoma civilizovan mesec, ali ne vidimo ni jedan razlog da sledeći ne bude još bolji.

VICEVI

Željko Krstić

Mesec mart je protekao u znaku povećanog priliva poruka u konferenciju **VICEVI**. Bilo je tu dosta viceva koji su već viđeni, ali je za razliku od februara bilo i pregršt novih viceva. Primećen je veliki broj novih pitalica tipa „Kakva je razlika između ... i ...", „Šta je to ...". Neke od njih su upravo na glasanju za vic meseca. Kad već pomenusmo vic meseca, da kažemo da je vic februara bio jedan „old tajmer" koji se bavi sada već gotovo zaboravljenom Olimpijadom u Sarajevu. Na svečanom otvaranju se, kao što i pristoji, okupio silan narod, sva mesta zauzela. Konferansje počinje: „Olimpijske igre u Sarajevu otvoriće drug Mujo!" Iz publike se čuje mlakušan aplauz. Drug Mujo izlazi, staje pred mikrofon, vadi papir, čita: „Drugarice i drugovi ..." Iz publike se čuje slabašan aplauz. „OOOO!" Iz publike se čuje snažno „OOOO" „OOOOOOOOO!" Iz publike oduševljeno ceo stadion u jedan glas: „OOOOOOO!" „OOOOOOOOOOOOO!" U publici delirijum, seku vene, skaču pirama, govornicu lete kape i rublje, ekstaza: „OOOOOOOOOOOOOOOOO!" Konferansje šapuće drugu Muji: „Čitaj dalje, to su bre olimpijski krugovi!"

SezamFile 14

Jovan Bulajić

I u mesecu aprilu na Sezamu je bilo novih programa u izobilju – više od 10 megabajta u devedesetak datoteka. Među njima je bilo nekoliko kompletnih velikih aplikacija, odličnih *utility* programa, a najtraženije su bile dva domaće igre. Otvorena su i još dva nova direktorijuma – DATABASE (za baze podataka) i WORDPROC (za obradu teksta).

Predstavićemo vam najtraženije programe u aprilu, nekoliko pravih velikih *shareware* aplikacija, malo sistemskih alatki i dva odlična domaća *utility* programa.

Najtraženiji programi

Ovaj put su najtraženije datoteke bile dve domaće igre – nova verzija *ZB Tetrisa* (autori su **Dragan Zakić** i **Dejan Rudić**) i još jedna avantura inspirisana Sezamom – PAUS ili Misterije spasa (autori su **Srđan Veljković** i **Darko Milutinović**). Od "konsnijih" uslužnih programa, korisnici Sezama su najviše interesovanja pokazali za PKZip 2.04g (koji već mesecima ne silazi sa vrha top-liste *download*-a) i *McAfee Scan & Clean* paket za borbu protiv virusa.

Nadležne državne komisije su nedavno usvojile novi standard za slova srpske ćirilice i latinice na računarima: JUS I B1 015 (ovaj standard su korisnici i programeri na Sezamu dosta kritikovali, ali nećemo ovde pričati o tome). Već je razvijen i program (drajver) za prikaz znakova po novom standardu (za VGA kartice) i njihovo unošenje sa tastature. Program se zove All4Yu (pretpostavljamo da naziv potiče od toga što sadrži sve što je potrebno za rad sa YU slovima po tom standardu); nosilac projekta bila je firma IMTEL, a autor je **Nenad Bogojević** (*nkbog* na Sezamu) (**\ibmpc\yu\ jus_char.zip**)

Nova verzija programa *VPIC* je takođe izazvala dosta interesovanja među Sezamovcima. *VPIC* je jedan od brojnih programa za pregled slika i konverziju grafičkih datoteka. Radi na EGA, VGA i Super VGA video karticama, na žalost *ne* i na Herculesu. Od Super VGA kartica podržane su sve značajnije – Tseng, Trident, Oak, Paradise, Western Digital, S3, 8514A. Slike koje se mogu pregledati i konvertovati mogu biti u jednom od sledećih formata: BIF (Binary Image Format), BMP (Microsoft Windows 3.x Bitmap), GIF (GIF87a i GIF89a standard), LBM (Deluxe Paint), MAC, PIC (Pictor/PC Paint, ViewPoint), PCX (PC Paintbrush), CUT (Dr Halo), SCx (Colorix EGA Paint) i TGA (Targa, i to format od 8 do 32 bita po pikselu) (**\ibmpc\graphics\ vpic60c.arj – a01**)

System

Tokom aprila "McAfee" nije izdao nove verzije svojih antivirusnih programa, ali nas je iznenadio programom za pregled i analizu sistemskih podataka. *ProView 1.2* služi za dinamički pregled i menjanje osnovnih komponenti računarskog sistema – kao što su sistemske promenljive, sistemska memorija, interapti, *device* drajveri, flopi i hard diskovi, datoteke. Podaci se mogu pregledati u ASCII, HEX ili u obliku disasembliranog koda.

Ova alatka može biti veoma korisna za pregled stanja sistema na najnižem nivou; između ostalog može se primeniti i za otkrivanje nepoznatih virusa praćenjem onoga što rade potencijalno opasni rezidentni programi. Ovo se lako može utvrditi (naravno potrebno je dosta znanja o strukturi MS-DOS sistema) ako proverimo da li je neki nepoznati program prisutan kao rezidentan i "zakačen" za neku od interapt funkcija. Šteta što *ProView* ne može da radi kao rezidentni program, jer bi onda bio mnogo korisniji, ali i ovako ga vredi pogledati (**\ibmpc\system\ pv12.zip**)

Korisnici DR-DOS-a (još jedne zamene za standardni MS-DOS komandni procesor, proizvod firme "Digital Research") bi trebalo da pokupe *update* za *utility* programe za verziju DR-DOS 6.0, datiran novembra '92. Paket sadrži nove verzije datoteka backup.com, command.com, cursor.exe, diskcopy.com, fdisk.com, format.com, keyb.com i restore.com, dakle vitalne i često korišćene delove sistema (**\ibmpc\system\ util1192.zip**)

Izašla je i nova verzija komandnog procesora 4DOS. U verziji 4.02 nije bilo previše izmena, i uglavnom su se ticale kompatibilnosti sa novim MS-DOS-om 6.0 (**\ibmpc\system\ 4dos402*.***)

Veliki programi

Glavna odlika većine programa koji se distribuiraju kao *shareware* (pod ovim ubrajamo i *public domain*, *freeware* i slične termine) je da su mali i imaju relativno usko područje primene (npr. razni uslužni programi koji rade samo jednu stvar i ništa više).

Međutim, među *shareware* programima ima i kompletnih aplikacija koje ni po čemu ne zaostaju za onima koje spadaju u komercijalne. Njihovi autori su uložili u to mesece, pa čak i godine rada, a zaključili su da *shareware* konceptom mogu ostvariti više koristi nego da programe distribuiraju kao komercijalne.

Među aprilskim novitetima na Sezamu se našlo i nekoliko programa koji bi se mogli svesti pod "velike aplikacije" – po jedan tekst procesor, baza podataka, zatim programi za tabelarne proračune (spreadsheet), stono izdavaštvo (DTP) i projektovanje & dizajn (CAD).

EnVision Publisher je kompletan program za stono izdavaštvo (*desktop publishing*). Radi u sopstvenoj grafičkoj okolini (nas podseća na GEM Venturu) uz potpuni WYSIWYG i korišćenje miša. Koristi vektorske fontove i, naravno, ima sve osnovne mogućnosti kao i komercijalni DTP sistemi: rad na svim grafičkim karticama i podržava sve vrste štampača. Sigurno da nedostaju razne stvari koje služe za pravo profesionalno korišćenje, ali za manje zahteve (npr. pravljenje raznih biltena) *EnVision Publisher* je više nego dovoljan (**\ibmpc\wordproc\ evp10.arj – a03**)

Poseban kuriozitet među novitetima u aprilu su program *PC File* i *PC Write*. Autor *PC File*-a, Jim Button, se smatra začetnikom *shareware* koncepta (uz Andrew Fleugelmana, autora PC Talk-a), dok je *PC Write* (autor je Bob Wallace) treći u tom istorijskom redosledu. Za proteklih desetak godina oba programa su se značajno razvila i donela njihovim autorima dosta koristi. Po podacima sa kraja prošle decenije (na žalost, ne raspolažemo novijima, ove smo preuzeli iz knjige "Dvorak's Guide to PC Telecommunications"), kompanija "ButtonWare" je imala godišnje prodaje od preko 2.5 miliona USD, dok je *PC Write* imao više od 45 000 registrovanih korisnika (ove cifre su danas sigurno mnogo veće).

Današnja verzija programa *PC-File* je solidan sistem za rad sa bazama podataka; koristi popularni DBF format datoteka (*dBase, Clipper, FoxBase/Pro*) i sopstvenu grafičku okolinu koja veoma lepo izgleda, ali je, na žalost, dosta spora na slabijim računarima. Čini nam se da je za korisnički interfejs korišćena *Zinc* biblioteka, ista ona koju koristi i "PC-Magazine" za svoje *benchmark* testove (**\ibmpc\database\ pcf65.arj – a04**)

PC Write je tekst-procesor u rangu sa najboljim DOS programima za obradu teksta. Na Sezamu se nalazi verzija *PC Write Advanced Level 4.01*, a postoji i *Lite* verzija, za koju kažu da je jedan od najboljih izbora za vlasnike prenosnih računara manjih mogućnosti (**\ibmpc\wordproc\ pcw401.arj – a04**)

Preostale dve noviteta iz ove klase velikih aplikacija su *spreadsheet* program *As Easy As v5.50c* (klon programa *Lotus 1-2-3*, **\ibmpc\misc\ asa55c.arj – a02**) i *Draft Choice v2.00b* (program za projektovanje i tehničko crtanje, **\ibmpc\graphics\ dc20b.arj – a01**)

Utility

Na ovim prostorima nekada je bio vrlo raširen mali rezidentni program koji je prikazivao tačno vreme u gornjem desnom delu ekrana. Čini nam se da je poreklo vodio iz nekog starog PC Magazine-a i bio je popularan uprkos brojnim manama, kao što je nemogućnost podešavanja pozicije na ekranu, problemi sa skrolovanjem i slično.

Ali, sada smo dobili jedan domaći program iste namene koji ima sve potrebne opcije, pa i više od toga. *ZTime v1.3* je rezidentni sat kod koga se mogu podešavati pozicija, boja ispisa, način prikazivanja. Može se postaviti alarm (samo jedan, ali to je obično dovoljno), pa će vas *ZKey* upozoriti u određeno vreme. Vrlo korisna opcija je to što satić ne mora biti stalno prisutan na ekranu, već se može aktivirati pritiskom na *hot key* kombinaciju, a zatim se posle izvesnog vremena opet skloniti sa ekrana. Kao rezidentan zauzima svega 960 bajta, tako da nećete mnogo izgubiti ako ga instalirate u autoexec. Autor *ZTime*-a je **Dragan Zakić** (dzakic) (**\ibmpc\utility\ ztime14.zip**)

FileFinder-a popularnih programa za traženje datoteka po disku ima zaista mnogo, ali nam se čini da je jedan od najboljih *BFind*. Autor programa je **Bojan Petrović** (*pbojan*), a u aprilu je napisao verziju 1.2.

BFind je pre svega brz – u najnovijoj verziji brži od svih nama poznatih *file finder*-a. Ali njegovu upotrebljivost čini pre svega veliki broj opcija i mogućnosti koje poseduje – npr. pretraživanje po regularnim izrazima (znatno složeniji i fleksibilniji sistem od krutih DOS-ovih džoker znakova * i ?), unutar arhiva (ZIP, ARJ i ostale), filtriranje po datumskom opsegu, veličini datoteke, atributima. Nad datotekama koje zadovoljavaju uslove mogu se izvršavati DOS ili druge komande, ili se tražiti stringovi unutar njih. Naravno, *BFind* poseduje i vrlo korisnu mogućnost traženja duplikanih datoteka po disku.

Evo nekoliko primera korišćenja – npr. sve datoteke ekstenzije BAK sa diska možete elegantno obrisati sa **BFIND *.bak /del** ili napraviti arhivirani bekap svih C programa menjanih od 1. maja sa **BFIND *.c *.h /s010593 /com"pkzip backup.c &f"**

Autor *BFind*-a daje primer iz kojeg najbolje možete videti kako se sve mogu filtrirati datoteke:

BFIND cd /grep t~[agr1-5]* ?x?" /gt1024 /s010193 /arh /etta /zip

Ovo znači: nađi na diskovima C i D sve datoteke koje počinju slovom "T", negde u imenu imaju A, G, R ili broj između 1 i 5, srednje slovo ekstenzije im je "X", veće su od 1024 bajta, kreirane su posle 1. januara 1993. godine, setovan im je arhiv bit, a pri svemu tome pogledaj i u arhive, i to samo ZIP! (**\ibmpc\utility\ bfind12.arj**)

Tabela 1 – TOP 10 datoteka – april 1993

	direktorijum	datoteka	opis	
1.	\IBMPC\FUN	zbtet30	zip	ZBtet 3.0 – domaći tetris (dzakic & beast)
2.	\IBMPC\FUN	paus101	arj	PauksA u Sezamu-avantura na SUPI (Veljković, Milutinović)
3.	\IBMPC\ARCHIVER	pkz204g	exe	PKZip arhiver (v 2.04g)
4.	\IBMPC\VIRUS	scanv102	zip	McAfee SCAN 9.13v102 virus detektor
5.	\IBMPC\YU	jus_char	zip	VGA drajver za ćirilicu i latinicu po I.B1.015 (imtel)
6.	\IBMPC\CLIPPER	bumenu11	arj	BuMenu 1.1 – poboljšani sistem menija za Clipper (bulaja)
7.	\IBMPC\VIRUS	clean102	zip	McAfee Clean 9.13V102 – uklanja viruse koje SCAN nađe
8.	\IBMPC\WORDPROC	evp10	arj	EnVision Publisher – program za stono izdavaštvo (DTP)
9.	\IBMPC\GRAPHICS	vpic60c	arj	VPIC v6.0c – pregled i konverzija slika (EGA/VGA+)
10.	\IBMPC\SYSTEM	pv12	zip	ProView 1.2 – pregled raznih sistemskih podataka (McAfee)

NAJVEĆA NAGRADNA IGRA U ISTORIJI ČASOPISA RAČUNARI

40.000 DEM U HARDVERU I SOFTVERU

JEDINSTVENA PRILIKA DA SPOJITE LEPO SA KORISNIM

I DOĐETE DO JEDNE OD PREKO 100 VREDNIH NAGRADA OD KUTIJE DISKETA DO 486/50 MHz SA LOKAL BASOM

• Velika nagradna igra u četiri broja „Računara" • Preko 100 nagrada od čega 8 u vrednosti od preko 1000 DEM • Nagradni kupon u svakom broju • Kuponi se šalju na kraju igre • Javno izvlačenje do kraja oktobra • Uručenje nagrada odmah po izvlačenju • Rezultati igre u decembarskom broju

U saradnji sa nekoliko vodećih domaćih firmi za promet i razvoj računske opreme redakcija „Računara" po tradiciji za ovo doba godine pokreće veliku nagradnu igru za čitaoce koja će trajati celog leta. Za razliku od dosadašnjih igara, od vas ovoga puta ne tražimo ništa posebno – dovoljno je da sakupite kupone sa zaštitnim znakovima naših sponzora i dostavite ih, kada za to dođe čas, na našu adresu.

Igra će se odvijati sledećim redom:

- ***računari*** 92 najava nagradne igre + COMTRAD kupon
- ***računari*** 93 predstavljanje sponzora i nagrada + INFO D kupon
- ***računari*** 94 predstavljanje sponzora i nagrada + PC PLUS kupon
- ***računari*** 95 predstavljanje sponzora i nagrada + IMTEL kupon
- ***računari*** 97 rezultati i izveštaj sa izvlačenja

NAGRADE

1. ComTrad	Računar 486DX / 50 MHz, local bus, 8 MB RAM, local bus video sa 1 MB RAM-a, hard disk 210 MB, FDD (5.25", 3.5"), SVGA monitor (Daewoo), veliko tower kućište, tastatura, miš PLUS, [illegible] Hewlett Packard DESKJET 500
2. IMTEL	**Specijalna novčana nagrada**, garantno pismo u vrednosti od **5.000 DM** za kupovinu opreme po sopstvenom izboru
3. Info D	Računar 486DX / 50 MHz, ISA, 4 MB RAM, Trident 1 MB, hard disk 210 MB, FDD (5.25", 3.5"), SVGA color (Philips) monitor, veliko tower kućište, tastatura, miš
4. PC Plus	Računar 386DX / 40 MHz, 4 MB RAM, VGA 512 KB, hard disk 130 MB (Seagate ST3144A), mono VGA monitor (RIC), FDD (5.25", 3.5"), fax/modem kartica, SoundBlaster „Zoltrix" sa zvučnicima
5. Jugodata	Računar 386DX / 40 MHz, Trident (8900), 4 MB RAM, hard disk 80 MB, mono VGA (SCAN), mini tower kućište, tastatura, miš
6. LS Data	Džepni kompjuter PSION series 3, 256 KB RAM
7. Adacom	Štampač PANASONIC KX P1123 (24 iglični)
8. SoftProjekt	Štampač PANASONIC KX P1123 (24 iglični)
9. Računari	Štampač CITIZEN 124 D (24 iglični)
10. SAGA	Štampač Star LC 20 (9 iglični)
11. MR Sistem	Štampač HYNDAI HDP-920 (A3 9-iglični)
12. ISC	Štampač HYNDAI HDP 920 (A3 9-iglični)
13. ASYS comerce	Štampač EPSON LX 100 (9 iglični)
14. OSA	Video VGA kartica (Truevision 16F PAL) (OSA). PAL grafička mašina za konverziju VGA u video do rezolucije 800x600 u 256 boja i VGA prikaz do 1024x768 u 256 boja. Omogućuje snimanje VGA grafike na video, kao i njeno spajanje sa živom slikom
15. MICRO AIR	Grafička kartica DIAMOND Stealth 24 (16.7 miliona boja), S3 grafički čip, res. 1280x1024 sa 1 MB RAM i prateći softver za podršku velikog broja aplikacija
16. MICRO AIR	Grafička kartica DIAMOND SpeedSTAR HI color (32.000 boja), TSENG grafički čip, res. 1024x768 sa 1 MB RAM i prateći softver za podršku velikog broja aplikacija
17. Micoudata	On line UPS uređaj od 500 VA sa petnaestominutnom autonomijom
18. KR DATA	UPS – 750, neprekidno napajanje za desktop računare 750 W
19. MP Biro	Color VGA monitor (DTK 14")

PROPOZICIJE

1. Pravo učešća, mogućnost dobijanja neke od nagrada imaju svi čitaoci „Računara". U nagradnoj igri ne mogu da učestvuju zaposleni u firmama sponzorima i u „Računarima", odnosno BIGZ-u, članovi njihovih porodica, kao i lica koja su redovni saradnici organizatora igre.
2. Nagradni kuponi objavljuju se u četiri broja u „Računarima": broj 92 (maj 1993.), broj 93 (jun-avgust 1993.), broj 94 (septembar 1993.) i broj 95 (oktobar 1993.).
3. Učesnici treba da pošalju sva četiri nagradna kupona. Za dodelu nagrada u obzir će se uzimati jedino originalni nagradni kuponi iz „Računara".
4. Sva četiri nagradna kupona treba – isključivo zajedno – dostaviti na adresu: „RAČUNARI", Bulevar Vojvode Mišića 17, 11000 Beograd, sa obaveznom naznakom NAGRADNA IGRA. Rok za dostavu: **25. oktobar** (uvažava se poštanski žig).
5. Kuponi mogu da se šalju isključivo u standardnim kovertima manjeg ili srednjeg formata. U obzir za dodelu nagrada neće se uzimati veliki koverti (formata A3 ili veći), koverti pravljeni u vlastitoj režiji, paketići i slično.
6. Redakcija „Računara" i sponzori kao organizatori ne mogu da snose nikakvu odgovornost za eventualni gubitak, oštećenje ili kašnjenje vašeg pisma u pošti.
7. Izvlačenje imena dobitnika nagrada izvršiće se javno, najkasnije do kraja oktobra. O tačnom datumu i mestu izvlačenja učesnike ćemo obavestiti preko sistema za modemske komunikacije Sezam.
8. Nagrade će biti uručene dobitnicima odmah po izvlačenju.
9. Kompletni rezultati izvlačenja će biti objavljeni u decembarskom broju (97) „Računara".

20. Tehnicom Panasonic telefon

21. Softverski paketi:

- dBASE IV **(Sprint)**
- Turbo Pascal for Windows **(Sprint)**
- Turbo Pascal 6.0 DOS ver. **(Sprint)**
- Quattro pro 4.0 **(Sprint)**
- Autodesk – CA LAB (dva kom.) **(OSA)**
- Autodesk – CHAOS (dva kom.) **(OSA)**
- MS DOS 6.0 Upgrade **(MISoft)**
- Borland Quattro pro 1.0 **(Softland)**

22. Kursevi

- Kurs po izboru (za deset dobitnika) **(Tehnicom)**
- AUTOCAD ili COREL DRAW (za šest dobitnika) **(Micro Animo)**
- Kurs po izboru (za pet dobitnika) **(Institut "Boris Kidrič")**

23. Miševi

5 komada	**Sprint**
2 komad	**Tehnicom**
1 komad	**MP Biro**

24. Diskete

8 kutija	SAGA
2 kutije –	Tehnicom

25. Filter za monitor

5 komada	Sprint

26. Knjige

10 knjiga –	Institut „Boris Kidrič"
1 knjiga	CET

27. Držač za papir i filter CET

28. Podsetnik za korišćenje tastature za WP 5.1 i Windows CET

29. Pretplata na Sezam

- 1 godišnja
- 2 polugodišnje
- 3 tromesečne
- 4 jednomesečne

30. Pretplata na časopis Računari

- 2 godišnje
- 2 polugodišnje
- 2 tromesečne

Info-D
informatički inženjering

NAGRADNI KUPON 2
računari 93
Sačuvajte kupon do kraja nagradne igre

SVI NAŠI SPONZORI





COMTRAD

Genex apartmani, V. Popovića 8, Beograd, tel: 222-41-51, 222-36-52, fax: 222-41-38.

COMTRAD Group je organizacija osnovana u Venecueli 1984. godine i bavi se proizvodnjom i prodajom informatičke opreme i usluga. Zapošljava više od 700 ljudi u oko 40 firmi širom sveta. Kompanija COMTRAD YU sa sedištem u Beogradu, pored jugoslovenskog, zadužena je i za istočno-evropsko tržište, a deo posla preuzima i sestrinska firma „SITI". Kao deo grupe, COMTRAD YU nudi PC AT računare 386SX, 386 i 486, „Epson" i „Hewlett Packard" štampače, „Novell" mrežni softver, multiportne kartice za UNIX, UPS uređaje, mrežnu opremu kao i sve vrste prateće opreme.

INFO-D

Gradski park 8 II (Zgrada Crvenog krsta), 11080 Zemun, tel./fax 612-822.

Agencija za informatički inženjering, marketing i promet INFO-D je osnovana 1986. godine. Bavi se razvojem softvera, kao i prodajom računara i pratećeg hardvera. Nudi komponente najpoznatijih svetskih proizvođača – Western Digital, Conner, Intel, Teac, Toshiba, Epson... U saradnji sa domaćim firmama INFO-D izrađuje standardni i namenski softver, a razvijen je i korisnički alat za domaće programere.



IMTEL

Bulevar Lenjina 165b, Novi Beograd, tel: 136-420, fax: 136-836

IMTEL, Institut za mikrotalasnu tehniku i elektroniku, je naučno-istraživačka kuća osnovana 1973. godine. IMTEL u svom računarskom tržišno orijentisanom programu nudi proizvodnju, prodaju i servisiranje računara i računarske opreme, prodaju, projektovanje, instalaciju i održavanje računarskih mreža, izradu programa vezanih za poslovanje firmi i preduzeća, računarski sistem za bespapirno poslovanje *IMTE-Loffice*, obuku – počevši od osnova računara do uske specijalizovanih programskih paketa, te stručne konsultacije. Preko javne informacione elektronske mreže IMTELBBS svaki korisnik IMTEL-ove računarske opreme može da pristupi IMTELNET-u koji je povezan sa JUPAK-om.



PC PLUS

Budimska 18a, 11000 Beograd, tel/fax: 337 126, 346-227

Firma PC PLUS je osnovana 1992. godine. Osnovna delatnost PC PLUS-a je projektovanje i instalacija računarskih mreža zasnovanih na Novell operativnom sistemu i programu Microsoft Windows for Workgroups, kao i izrada računarskih sistema svih konfiguracija sa pratećom opremom po želji korisnika. PC PLUS vrši kompletno servisiranje sistema i opreme. Kao ovlašćena firma za prodaju originalnog softvera najpoznatijih proizvođača, PC PLUS isporučuje najnovije programske pakete koji se trenutno nude i u svetu.

JUGODATA

Bulevar revolucije 326, 11000 Beograd, tel/fax 419-326, 419-766, (061) 11-618

JUGODATA projektuje računarske sisteme po zahtevu kupca. Isporučuje svu opremu, od najskromnijeg PC računara do mreže jakih računara i neophodne periferijske opreme, opremu sistemskim softverom iz svoje standardne ponude ili ga izrađuje po zahtevu kupca. Obezbeđuje obuku te servisiranje i održavanje računarskih sistema. Kao generalni zastupnik singapurske firme „S&C Computer", JUGODATA nudi i pojedinačnu prodaju računarskih komponenti, periferne opreme i potrošnog materijala. Ima brojne instalacije lokalnih mreža baziranih na Novell Netware sistemu i razvija aplikacije za ovakva višekorisnička okruženja.





MICRO DATA

Ustanička 64/XII, Beograd, tel/fax 454-891, 488-686

Preduzeće MICRODATA se bavi prodajom svih konfiguracija PC računara, sistema za neprekidno napajanje (UPS), "Epson" štampača i mikro-filmske opreme. Izrađuje softver za finansijsko poslovanje i knjigovodstvo te vrši projektovanje računarskih mreža i kompletno servisiranje računarskih sistema.



OSA

Šumadijski trg 5a, Beograd, tel 557-088, fax 559-845.

Osnovna delatnost je profesionalna računarska grafika u oblasti projektovanja na računaru (CAD) fotorealistična vizuelizacija prostornih modela i računarska animacija. Korisnicima računara OSA nudi savetovanje projektovanje sistema isporuku i umrežavanje računare periferijske uređaja standardni ili specijalni softver kao i obuku kadrova i održavanje sistema



SOFT PROJEKT

Vraneši, Vrnjačka Banja, tel 036/61-647 tel. 036/67-271

Preduzeće za informatički inženjering i trgovinu SOFT PRPJEKT iz Vrnjačke Banje postoji već 5 godina. Osnovna delatnost je izrada softvera po narudžbini a specijalnost knjigovodstveni softver koji se realizuje pod DOS-om i NOVELL-om Osim toga SOFT PROJEKT je i distributer računara i prateće opreme iako da na tržištu nastupa sa ponudom po principu „ključ u ruke"



ISC

ISC Lipa 35/117 Cerak, Beograd, tel/fax 506-785.

Preduzeće ISC se bavi konsaltingom u oblasti informacionih sistema i prodajom IBM kompatibilne opreme - računara AT/386 i 486 (ISA, EISA VL Bus) sa orijentacijom na jednoprocesorske i višeprocesorske servere grafičke stanice i mrežne terminale



MR SYSTEMS

MR Sistem Lenina 38, Beograd, tel: 000-000, 506-786, fax 667-285.

MR SYSTEMS je preduzeće za razvoj i izgradnju informacionih sistema koje posluje u zemlji i inostranstvu. Osnovna delatnost je inženjering u oblasti informacionih sistema za poslovnu i industrijsku primenu na UNIX i DOS platformi u TCP/IP mrežnom okruženju sa akcentom na CIM rešenjima.



SAGA

Saga Milentija Popovića 9 (Sava Centar), 11070 Novi Beograd, tel/fax: 222-36-78, 147 188, 222-43-23/lok 266, 268.

Osnovna delatnost SAGE je kompjuterski inženjering proizvodnje projektovanje i instalacija računarskih sistema i mreža, i druge prateće opreme SAGA je zastupnik američke kompanije ATHANA japanske firme STAR i američke firm „American Power Conversion" (APC)



ASYS COMMERCE

Bulevar 23. oktobra 25, Novi Sad, tel 021/623-588, 624-58 | fax. 021/615-867

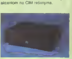
ASYS COMMERCE je preduzeće za računarski inženjering i konsalting U svojoj ponudi ima računarsku opremu različitih performansi i kapaciteta a posebno štampače (Epson) linijske štampače (Dataproducts) i uređaje za neprekidno napajanje (UPS) ASYS COMMERCE posvećuje posebnu pažnju razvoju sopstvene servisne službe



MP – BIRO

Beogradska 41 Beograd, tel 341-382, 466-361

MP BIRO je mešovito grčko-jugoslovensko preduzeće za inženjering i marketing u oblasti računarske tehnike i informatike odnosno biro za organizaciono-tehnološke poslove automatsku obradu podataka stono izdavaštvo i prodaju računara i prateće opreme Za tri godine postojanja MP-BIRO je razvio svoju servisnu mrežu u Beogradu Nišu, Podgorici Prištini



KR DATA

Kosovska 18a, Novi Sad, tel: 021/623-611 611-153, fax 021/28-322

Glavna delatnost preduzeća KR DATE je projektovanje i izrada uređaja za neprekidno napajanje (KR UPS) kao i projektovanje izrada, instalacije i održavanje profesionalnih sistema za ozvučavanje KR DATA se bavi i prodajom servisiranjem i održavanjem računara i računarske opreme sopravkom raznih elektronskih uređaja



ANIMA

29 novembra 71 Beograd, tel 343-040, 343-284.

ANIMA je preduzeće za računarski podržano arhitektonsko projektovanje predstavu i animaciju. Nudi konsalting iz oblasti nabavke i primene savremene računarske opreme u arhitektonskoj praksi kao i obuku kadrova Osim toga. ANIMA nudi iznajmljivanje visokokvalitetnog HP plotera laserskog i kolor štampača kolor skenera i slajd rekordera kao i prenošenje slike sa računara na video traku



SPRINT

Poslovni centar "Apolo" II sprat, Trg slobode 3, Novi Sad, tel. 613-800. 611-892, 623-717 fax 623-901

SPRINT INFORMACIONI SISTEMI je asocijacija zanatskih radnji koje su se integrisale pre četiri i po godine Bavi se sklapanjem održavanjem i servisiranjem računara, projektovanjem informacionih sistema, projektovanjem i izradom računarskih mreža razvojem aplikativnog softvera, nabavkom originalnog softvera i obukom korisnika.



MICRO-AIR

29. novembra 71, Beograd, tel: 343-040, 343-284.

Preduzeće MICRO-AIR se bavi proizvodnjom računara za CAD i grafičke aplikacije u raznim konfiguracijama. Zastupnik je nekoliko poznatih svetskih firmi i nudi njihovu računarsku opremu „Hewlett Packard" štampače skenere plotere i dodatnu opremu „Epson" štampače EIZO profesionalne monitore, „Diamond" i „Hercules" grafičke karte „Polaroid" film rekordere



LS DATA

Gračanička 8/1 tel/fax: 638-768, centrala 638-866/16

LS DATA distribuira PSION računare i pruža kompletan inženjering – izradu projekata instalaciju PSION opreme, povezivanje PSION opreme sa PC računarima izradu programa za PSION i PC-računare kao i obuku i servisiranje LS DATA prodaje i PC-računare štampače i ostalu računarsku opremu



CET

Studentska 46, Beograd, tel/fax 349-048.

Preduzeće CET je osnovano pre 3 godine i bavi se prodajom računara i različite računarske opreme – hardvera licencnog softvera potrošnog materijala, strane i domaće literature, kompjuterskog nameštaja. CET je ovlašćeni distributer više softverskih firmi, među kojima su i Borland i Microsoft



INSTITUT „VINČA"

Centar za permanentno obrazovanje

Nemanjina 4/10 Beograd, tel 683-388, 682-486. 641 155/lok 187 181 fax 682-486.

Bavi se obrazovnom i izdavačkom delatnošću u oblasti programiranja programskih jezika i primene računara, kao i izradom poslovnih i poslovno-tehničkih informacionih sistema. Jedini je autorizovani obrazovni centar u Srbiji za proizvode firmi SCO ORACLE BORLAND AUTODESK i LOTUS



ADACOM

Čika Ljubina 12, Beograd, tel: 638-233, fax 638- 672.

ADACOM se bavi sklapanjem i prodajom IBM-kompatibilnih računara od klase 386SX do 486 na 50MHz EISA Osim toga, nudi i svu prateću opremu za računare - štampače miševe strimere i druge ADACOM obavlja i projektovanje i instalaciju računarskih mreža za koje pruža potpunu softversku i hardversku podršku



TEHNIKOM

Vojvode Mišića 37 Beograd, tel/fax 236-11-66, tel 698-522/1692.

Preduzeće TEHNIKOM je osnovano 1991 godine i bavi se prodajom računara servisnim uslugama u oblasti elektronike izradom softvera, projektovanjem a posebno obukom. TEHNIKOM ima sopstveni školski centar za 6 polaznika u kojem organizuje kurseve za MS-DOS korisničke programe Novell i Unix operativne sisteme

R A Z L I K A U



Business ja svuda

Ako hoćete da ostavite svoj trag na svakom poslu koji radite - kakav god da vam je posao, kakva god da vam je firma idealni ultraportabl PC za vas je Olivetti Quaderno 33

- *Ide sa vama* 1,3 kg i polovinu A4 formata možete poneti bilo kuda. On je sa vama uvek i svugde, omogućujući 9 sati rada sa jednim punjenjem baterija. Iako je tako malen može se povezati sa drugim računarima, štampačem, faxom ili modemom.
- *Govori kao vi.* najmanja multimedija platforma na svetu nudi vam mogućnost snimanja i reprodukovanja digitalnog zvuka u trajanju do 90 minuta
- *Pametan je kao vi* integrisani trackball miš omogućuje vam trenutnu i jednostavnu vezu sa pre-instaliranim Windows-om i MS Works-om (Personal Information Manager) kao i sa DOS-om instaliranim u ROM-u.
- *Misli kao vi* do 12Mb RAM-a i 60Mb HDU akumulatorske baterije i 386SXLV mikorpocesor

Quaderno 33 u potpunosti afirmiše tradiciju Olivetti-jevog dizajna, osmišljen i sačinjen da sruši barijeru proste racionalnosti i postane moderan i poželjan koliko i praktičan

Bilo kuda da se krećete, možete ostaviti konkretan znak individualne produktivnosti Vaše

olivetti

OLIVETTI ENERGODATA

P. Toljatija 5 11070 N. Beograd Tel. (011) 698.512. 687 393 Fax (011) 695 912


Dodjite i utvrdite!
TEH NICOM


MICROSYS
WE MAKE NETWORKING SIMPLE
GODINA 11 U INFORMATICI
Beograd, Molerova 70
432-690, 430-059



United Software of Microsoft
MAY WINDOWS 93
FOR EASTERN EUROPE
Microsoft
MAY ACCESS 93
Microsoft
MAY FOX PRO 93
FOR WINDOWS
Microsoft
MAY VISUAL 93
BASIC 2.0
Microsoft
MAY WINDOWS 93
FOR WORKGROUPS
Microsoft
MAY EXCEL 93
Microsoft
MAY C/C++ 7.0 93
Microsoft
MAY MS-DOS 93
VERSION 6.0
Microsoft
11000 Beograd
Skadarska 45
tel 011/343-043
MISOFT OVLAŠĆENI DISTRIBUTER KORPORACIJE MICROSOFT
BORLAND
The Leader in Object–Oriented programming
PARADOX for WINDOWS
4.0
BORLAND PASCAL 7.0
dBASE IV
C++
11000 BEOGRAD
SKADARSKA 45
TEL. 343-043
SOFTLAND

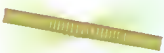


Nameštaj po vašoj meri!
it
Vojvode Mišića 37
11000 Beograd
tel/fax 235 1108
tel 650 522/1602

VREME JE...



ne bacajte ga uludo



U **CET Computer Shop** – u na jednom mestu ćete naći hardvere različitih proizvođača.

Mi smo ovlašćeni dileri svetskih proizvođača softvera **MICROSOFT, BORLAND, AUTODESK, LOTUS** ...

Posedujemo veliki izbor domaće i strane **literature**

Snabdevamo Vas, na Vaš zahtev, svim **potrošnim materijalom**.

Nudimo Vam **nameštaj** neophodan za formiranje računarskog radnog mesta

Naši eksperti pomoći će Vam da definišete svoje potrebe i realizujete svoje ciljeve

Computer **E**quipment & **T**rade
11000 Beograd • Skadarska 45 • Tel 343-043


ComTrad
the ART of
computer making
ENHANCED
POWER
FOR
WINDOWS
esor 386/40MHz, 4MB RAM
loppy uređaji 1 2MB i 1.44MB
Tvrdi disk 105MB IDE
Dva serijska i jedan paralelni port
SVGA graficki adapter 1024x768/1
SVGA kolor monitor
Mini tower kucište, ASCII tastatura, miš
2,290.-
ComTrad YU • Popovica 6, Beograd • (011) 222-41-51, 222-26-52 • Fax. (011) 222-41-39
SPRINT
Novi Sad
Novosadskog sajma 35
Tel. (021) 623-717
Fax (021) 623-901
INFOTRADE
Tel. (0 25-830
Fax (038)
MANIR
Ruma
Grobljanska 2
Tel (022) 421-265
Fax (022) 421-265
COMTRAD
KRAGUJEVAC
Kragujevac
Kragujevackog
oktobra 102
Tel (034) 510-929
Fax (034) 510-
ComTrad
COMPUTERS

BEST BUY

386 SX 33

1850 DM

2Mb expandable to 16Mb
1 44Mb floppy
105Mb (15mS) hard drive
512K SVGA card
SVGA monitor color 14"
DOS 6 0 + windows 3 1

2 godina garancija

BEST BUY

386 DX 40

2150 DM

4Mb expandable to 32Mb
128K cache. 1 44Mb floppy
130Mb (12mS) hard drive
1Mb SVGA card
AVGA3 16 7 milion colors
SVGA monitor color 14"
DOS 6 0 + windows 3 1

2 godine garancija

BEST BUY

486 DX 33

2650 DM

4Mb expandable to 32Mb
256K cache. 1 44Mb floppy
130Mb (12mS) hard drive
1Mb SVGA card
AVGA3 16 7 milion colors
SVGA monitor color 14"
DOS 6 0 + windows 3 1 + mouse

2 godine garancija

BEST BUY

486 DX 50

3050 DM

4Mb expandable to 32Mb
256K cache, 1 44Mb floppy
130Mb (12mS) hard drive
1Mb SVGA card
AVGA3 16 7 milion colors
SVGA monitor color 14"
DOS 6 0 + windows 3 1 + mouse

2 godine garancija

COMPONENTS

BOARD 386 DX 40	330 DM
BOARD 486 DX 33	900 DM
BOARD 486 DX 50	1250 DM
HDD "W DIGITAL" 130Mb	450 DM
HDD "W DIGITAL" 220Mb	900 DM
HDD "W DIGITAL" 255Mb	680 DM
FAX/MODEM CARD INTERNAL	280 DM
1 Mb SIMM	70 DM
SOUND BLASTER CARD	400 DM
MONITOR COLOR 14"	600 DM
MOUSE GENIUS "CLX PROFY"	80 DM
EPSON LX400 YU SET	480 DM
EPSON LQ100 YU SET	600 DM
EPSON LQ170	1400 DM
HELWETT PACKARD IIIp	2500 DM

MP-biro PREDNOSTI

Visoki kvalitet, najniže cene i mnoge dodatne povoljnosti:

* isporuka opreme na kuću
* ugradnja dodatne opreme
* mogućnost kasnije dogradnje sistema
* sastavljanje računarske opreme prema vašim potrebama
* programska podrška

WHAT PC?
PC Direct
PC
COMPUTER BUYER
NOVELL
MICROSOFT WINDOWS
SOUND BLASTER
Authorised dealer

MP BIRO

COMPUTER SYSTEMS

BEOGRADSKA 41
TEL. 011/341-392
496-351

NE DOZVOLITE DA VAS KONKURENCIJA PRETEKNE!

IMTEL OFFICE SYSTEM

POVEĆAJTE PRODUKTIVNOST!
PREĐITE NA BESPAPIRNO POSLOVANJE!
STVORITE NOVI STIL KOMPANIJI!

IMTEL OFFICE SOFTWARE

- INTEGRISANO WINDOWS OKRUŽENJE SA TEXT PROCESOROM, GRAFIKOM I SPREADSHEET PROGRAMOM.
- SISTEM ZA DISTRIBUCIJU DOKUMENATA PUTEM MREŽA (NOVELL, BRIDŽOVI, RUTERI)
- INTEGRISANI SISTEM ELEKTRONSKE POŠTE NA SVETSKOM NIVOU
- IZRADA INFORMACIONIH SISTEMA ZA PREDUZEĆA I FIRME.

- LOKALNE RAČUNARSKE MREŽE
- AUDIO I VIDEO MULTIMEDIJA
- ELEKTRONSKA POŠTA
- FAX SERVIS NA LOKALNOJ MREŽI
- POVEZIVANJE LOKALNIH MREŽA
- INTEGRISANE USLUGE PRENOSA TEKSTA, GLASA I SLIKE
- RAZMENA PODATAKA PREKO X.25 I X.400 PROTOKOLA SA KOMPANIJAMA U JUGOSLAVIJI I SVETU
- ULAZNO-IZLAZNI UREĐAJI U KOLOR TEHNOLOGIJI
- POVEZIVANJE UNIX, NOVELL I IBM MAINFRAME RAČUNARA
- APLIKATIVNI POSLOVNI SOFTVER ZA DOS I UNIX

INSTITUT ZA MIKROTALASNU TEHNIKU I ELEKTRONIKU
BULEVAR LENJINA 165b, NOVI BEOGRAD TEL 135428 FAX 138928

IMTEL

3.7.3 Komanda Snap to Guidelines



Slika 3.51

Ovom komandom takođe možete da poravnavate vaše objekte. To su pomoćne isprekidane linije (ne štampaju se i ne vide se u **Preview** režimu) koje se mogu postavljati uz pomoć **Guidelines** dijalog-boksa (slika 3.52), ili ih možete aktivirati tako da strelicu miša dovedete u prostor lenjira (horizontalni ili vertikalni), kliknete i zadržite pritisnut taster miša povlačeći ga do željenog mesta. Kada je komanda **Snap To Guidelines** aktivirana ove linije na sebe „privlače" objekte koje crtamo ili pomeramo. Udaljenost na kojoj će **Guidelines** početi da „privlači" objekte definiše se kroz komande u meniju **Special**, koji će biti naknadno objašnjen.

3.7.4 Komanda Guidelines Setup



Slika 3.53

Ova komanda omogućuje da uz pomoć **Guidelines** dijalog-boksa (slika 3.52) postavljate, pomerate, brišete i odabirete pojedine horizontalne i vertikalne pomoćne linije. Dijalog boks je jednostavan za upotrebu i biće u kratko opisan. Kod postavljanja pomoćnih linija iz ovog menija, potrebno je da odlučite da li postavljate horizontalnu ili vertikalnu liniju. Potom postavljate vrednost za njen položaj u odnosu na postavljeni **Grid origin**. I ovde je šematski prikazan koordinatni sistem sa predznacima kvadranata. Nakon što ste postavili vrednost odabirate odgovarajući taster:

Delete – briše odabranu horizontalnu ili vertikalnu pomoćnu liniju.

Move pomera odabranu pomoćnu liniju na položaj koji je definisan vrednošću u **Ruler Position**.

Add na definisanu poziciju dodaje novu pomoćnu liniju.

Next Pomoću ovog tastera možete da se šetate od jedne do druge pomoćne linije i aktivirate jednu od prethodno opisanih komandi tastera.

Close – Zatvara ovaj dijalog-boks. Ovaj taster nije u vezi sa odrađivanjem svakog od pojedinih tastera.

U okviru dijalog-boksa možete definisati da li je komanda **Snap To Guidelines** uključena ili ne. Dijalog-boks za editovanje pomoćnih linija možete aktivirati i dvostrukim klikom na određenu pomoćnu liniju, a tada se u polju predviđenom za upis broja za položaj linije nalazi podatak o tačnoj trenutnoj poziciji odabrane linije.

3.7.5 Komanda Snap To Objects

Komanda **Snap To Objects**, slično komandi **Snap To Guidelines**, poravnava – privlači objekat koji crtate ili pomerate prema nekom već nacrtanom bliskom objektu.

3.7.6 Komanda Show Rulers

Ova komanda uključuje ili isključuje lenjire, opisane u 1.2.

3.7.7 Komanda Show Status Line

Ova komanda uključuje ili isključuje prikaz podataka u statusnoj liniji.

3.7.8 Komanda Show Color Palette

Ova komanda (slika 3.53) vam pruža mogućnost izbora jedne od opcija: paleta isključena, paleta uključena sa prikazom procesnih boja, i paleta uključena sa prikazom definisanih boja. Ova komanda predstavlja samo nastavak svega što je već rečeno o bojama u *Corelu*.

3.7.9 Komanda Edit Wireframe

Ovom komandom uključujemo ili isključujemo **Preview** režim rada *Corela*. Kada je komanda uključena, objekti su prikazani na radnom papiru samo okvirnim linijama jednake debljine. Kada je komanda isključena, prikaz objekata je isti kao i prilikom aktiviranja komande **Preview** (što znači da se vide boje površine, boje i debljine linija). Ova mogućnost je verovatno ostavljena za neku vezu sa prethodnim verzijom *Corela* i radi lakšeg crtanja nekih objekata (pogotovo objekata sa debljim linijama, kada zbog debljine linija možete pogrešno da postavite određene objekte). Prethodne verzije zbog znatno manje brzine rada, nisu imale svu mogućnost da u radnom prostoru vidite stvarni izgled vašeg crteža. Postojali su razni modaliteti prikaza radnog i **Preview** prostora. Sve komande koje su vezane za ovakav način rada su u ovoj verziji izostale.

3.10 Ostali programi

Uz program *CorelDraw*, dobijete i neke druge programe koje kroz *Custom* instalaciju možete da instalirate. Ukoliko ste to uradili, u okviru programske grupe *Corel Graphics* će vam se pojaviti ovi programi:

CorelChart – program za grafički prikaz vaših podataka.

CorelPhoto-Paint – program za rad sa bitmapiranim slikama.

CorelMosaic – program za pregled i pretraživanje biblioteka sličica.

Corel Trace – program za transformaciju slika iz bitmapiranog formata u oblik krivih linija.

CorelCapture – program za „hvatanje" ekrana.

Miroslav Jovanović



CORELDRAW

KOMPLETNO UPUTSTVO ZA UPOTREBU

SPECIJALAN DODATAK ČASOPISA računari
jun-avgust 1993

1. UVOD

Sve veći zahtevi korisnika PC-ja naterali su proizvođače softvera da stalno rade na usavršavanju svojih programa. Program CorelDraw uza dešto prutalica Ponalo je nezahvalno pisati uputstvo za već postojeći softver kada se može očekivati da u međuvremenu izkrsne i nova verzija programa. Imajući sve u vidu pišem ove redove kako bi barem jedan deo zanesenih crtača mogao da uživa u svim blagodetima koje im pruža nova verzija CorelDraw 3.0

Sam program sadrži čitav niz sličnosti i razlika sa ostalim programima za crtanje. Za one koji su do sada već radili sa grafikom ovaj program će biti vrlo dobra alatka za rešavanje pojedinih problema, dok će onima koji do sada nisu radili grafiku otvoriti novi pogled na svet dizajna.

Najnovija verzija *CorelDraw 3.0*, pored programskih aplikacija koje su imale i prethodne verzije (*Mosaic Corel Trace*) ima i nove: *CorelChart*, *CorelPhoto-Paint* i *CorelShow*. O svakoj od navedenih aplikacija biće nešto kraće reči i u ovom tekstu, ali praktično svaka od njih zaslužuje i detaljnije uputstvo.

1.1 Instalacija i pokretanje

Za instaliranje paketa *CorelDraw* potrebno je prethodno imati instaliran i pokrenut *Microsoft Windows* (najbolje najnoviju verziju). Procedura instalacije je sledeća: iz **Program Manager**-a pokrenite komandu **Run** iz menija **File.** Disketu br 1 ubacite u disk-drajv (A ili B) i nakon toga otkucajte **A:Install (Enter)**. Dalje se prepustite programu, kojem ćete samo morati davati podatke gde želite da smestite vaš *Corel*, primere slična i drugo. Nakon ispravne instalacije, na *Windows* ekranu će vam se pojaviti programska grupa **Corel graphics** (slika 1.1)



Slika 1.1

Sam *Corel* pokrećete duplim klikom na ikonicu **Coreldrw**. Nakon aktiviranja programa, Corel će vas pozdraviti stranicom kao na slici 1.2. Najveći deo ekrana zauzima prozor za crtanje (Drawing Window). Na njemu možete da crtate i pišete gde god želite ali samo ono što se nalazi na stranici za crtanje (Printable page) biće moguće kasnije i štampati. Stranica za crtanje predstavlja vaš pravi radni prostor. Ona je predstavljena osenčenim pravougaonikom. Prozor za crtanje je uokviren lenjirima (Rulers), statusnom linijom (Status line), alatkama (Toolbox), linijom menija (Menu bar), naslovnom linijom (Title bar), paletom boja (Color palette), horizontalnom i vertikalnom linijom za skrolovanje (Scroll bar), i „tasterima" Restore, Minimize i Maximize. U samom radu ćete se sretati i sa mnogim dijalog boksovima. Izraz „dijalog boks" je teško prevesti. Može se opisati kao meni u obliku novog prozora u vašem programu, koji nudi čitav niz opcija za definisanje parametara aktivirane komande.



1.2 Lenjiri

Postavljeni su uz gornju i levu ivicu prozora za crtanje i koriste se za pozicioniranje objekata i definisanje njihove stvarne veličine. Lenjiri se mogu uključiti i isključiti komandama koje će biti kasnije objašnjene kroz opis menija Display (3.7)

1.3 Statusna linija

Smeštena je između linije menija i lenjira. Može se nazvati i linijom stanja, a u njoj se ispisuju podaci o odabranom objektu ili komandi koja se trenutno izvršava. Ona se takođe može uključiti i isključiti kroz meni **Display** (3.7)

1.4 Alatke

Na levoj strani ekrana nalazi se devet tastera sa devet različitih „alatki"

VELIKA STRELICA	**Pick Tool**
MALA STRELICA	**Shape Tool**
LUPA	**Zoom Tool**
OLOVKA	**Pencil Tool**
KVADRAT	**Rectangle Tool**
ELIPSA	**Ellipse Tool**
SLOVO	**Text Tool**
PERO	**Outline Pen Tool**
BOJA POVRŠINE	**Fill Tool**

O svakoj od ovih alatki biće više rečeno u daljem tekstu

1.3 Linija menija

File Edit Transform Effects Text Arrange Display Special

Nalazi se između statusne linije i naslovne linije. Sadrži imena padajućih menija ili *pull-down* menija.

1.6 Naslovna linija

CorelDRAW! - UNTITLED.CDR

Naslovna linija sadrži naslov *CorelDraw*, ime fajla u kojem radite i tastere za format prikaza *Corel* programa. U okviru ove linije nalaze se padajući meni (**Control menu box**), taster za uvećanje, taster za umanjenje i dugme kojim napravljene promene u veličini prikaza vašeg programa vraćate na prethodnu vrednost (**Restore**). *Windows* je i u ovom programu omogućio da se mišem obavite veći deo posla. Kada dođete do pojedinih ivica i odnosno uglova, kursor se promeni u strelicu koja može da menja dimenzije osnovnog ekrana *Corela*.

1.6.1 Padajući meni

Pored naredbi koje su identične onima koje izvršavaju tasteri (za redefinisanje veličina – format prikaza radnog prostora programa) tu se nalaze naredbe **Close** (zatvori), **Switch to** (pređi u). Naredba **Close** zatvara *CorelDraw* i pita vas da li želite da snimite crtež koji ste radili ili ne, samo pod uslovom da ga niste snimili posle urađenih izmena. Ova naredba je identična dvostrukom kliku na ikonicu ovog menija. Naredba **Switch** omogućuje da pomoću novog dijalog **boksa** pređete u neki drugi program (*Winword*, *Paintbrush* i dr.), pod uslovom da je on već prethodno aktiviran. Ukoliko nije aktiviran ni jedan program, onda vam se nudi mogućnost da pređete samo u **Program Manager**. Ukoliko vam je istovremeno aktiviran neki drugi program, dovoljno je da nakon otvaranja ovog dijalog-boksa odaberete program u koji želite da pređete i da u dijalog-boksu odaberete taster **Switch to**. U okviru ovog menija možete uz pomoć pojedinih naredbi da zatvorite programe koji vam više nisu potrebni (**End task**), da na određeni način složite aktivirane programe tako da ih sve istovremeno vidite (**Tile, Cascade**), da jednostavno odbijete da radite sa bilo kojom komandom (**Cancel**) i da složite ikone. Za jednostavniji prelaz iz jednog u drugi program češće ćete koristiti kombinaciju tastera **Alt+Tab**, koji će vas uz pomoć malog prozora voditi u program koji izaberete.

1.6.2 Taster za uvećanje

Ovim tasterom možete da postavite početni ekran vašeg programa tako da on zauzme maksimalan mogući prostor na ekranu. Identična naredba je **Maximize** iz prethodno opisanog menija.

1.6.3 Taster za umanjenje

Ovim tasterom postavljate veličinu vašeg programa na prethodnu – manju veličinu, a efekat je identičan naredbi **Minimize** iz prethodno opisanog padajućeg menija.

NAPOMENA: U ovom tekstu se neću zadržavati na opisu komandi koje su standardne za *Windows* okruženje. Neće biti posebno objašnjene radnje kao što su klik mišem, dvostruki klik, i slično. Detaljnija objašnjenja komandi karakterističnih za *Windows* okruženje se nalaze u okviru *Windows* help-programa ili odgovarajuće literature.

1.7 Pregledni ekran

Pregledni ekran (**Preview screen**) vam omogućuje da na formatu celog vašeg monitora vidite sliku koju ste napravili. Aktivira se sa funkcijskim tasterom F9, dok se pritiskom na bilo koje dugme vraćate u radni prostor. U preglednom ekranu ne možete raditi izmene crteža. Možete birati dve opcije: prikaz onoga što vam je u radnom prostoru ili prikaz samo selektovanih objekata.

1.8 Ivica radnog prostora

Ako imate uključenu naredbu **Maximize**, na ivicama će vam se nalaziti samo tasteri za pomak po prozoru za crtanje (*scroll bar*), dok se prave ivice prozora neće videti. One imaju ulogu samo u pravljenju manjeg prozora, u kojem ćete videti vaš radni ekran.

2. ALATKE

Na prethodnim stranicama rečeno je da na raspolaganju imate 9 kutija sa alatima, koje možete koristiti u ovom programu. U daljem tekstu će biti svaka od njih detaljnije objašnjena. Objašnjenja pojedinih alatki podrazumevaće da se već nalazite u određenom režimu rada, odnosno da ste vrh strelice miša postavili na ikonu (taster) alatke i klikom je aktivirali.

2.1 Velika sterlica – Pick Tool

Ova alatka se koristi za nekoliko načina izbora – selektovanja objekata. Potrebno je reći da sve što budete radili u *Corelu* počinje i završava objektom. Objekat predstavlja sve ono što zasebno crtate (linija, tačka, elipsa, slovo, pravougaonik i drugo), a čini ga skup čvorova povezanih linijama i površina koju ovi čvorovi zatvaraju. Bilo šta da želite da radite, morate prvo da odaberete objekat na kojem ćete raditi, pa tek onda možete primeniti neke od ponuđenih opcija iz ovog programa (pomeranje, promena veličine i drugo). Selektovan – izabran objekat možete prepoznati po tome što ga okružuje osam crnih kvadratića raspoređenih u obliku pravougaonika. Kada uđete u režim rada sa velikom strelicom (klikom miša na veliku strelicu) možete selektovati određeni objekat tako što vrh strelice miša dovedete na okvirnu liniju objekta (ukoliko je **Edit wireframe** u meniju **Display** isključen, može i na površinu objekta) i kliknete. Pojaviće se pomenutih osam crnih kvadratića. Selektovati možete i tako što postavite strelicu miša u okolinu objekta kojeg želite selektovati, kliknete mišem i držeći pritisnut taster miša, označite deo prostora koji sadrži objekte koje želite selektovati, te nakon toga otpustite dugme miša. Potvrda učinjenog posla biće pojava osam kvadratića. Na ovaj način možete selektovati više objekata zajedno. Ako već imate jedan objekat selektovan, do sledećeg možete stići uz pomoć tipke TAB. Tipka TAB vas vodi od objekta do objekta redosledom kojim ste ih crtali. Ako ste pravilno postupili prilikom selektovanja,

u statusnoj liniji ćete naći podatke o samom objektu (da li je to tekst, pravougaonik, elipsa ili nešto drugo), njegovoj okvirnoj liniji i boji površine. Naučite da se koristite informacijama iz statusne linije, to će vam mnogo olakšati pojedine poslove. Promenu odabranog – selektovanog objekta obavljate tako što kliknete na neki drugi objekat ili jednostavno kliknete u prostor između objekata, ako ne želite selekciju ni jednog objekta.

Svaki od kvadratića ima svoje značenje. Kada postavite miša na jedan od njih, strelica će se promeniti u krstić, koji vam omogućuje dalji rad sa objektom. Kvadratići na uglovima omogućuju vam da menjate dimenzije selektovanog objekta – kada se pojavi krstić, treba pritisnuti taster miša i tako pritisnutog pomerati po zamišljenoj dijagonali objekta do željene veličine. Veličina objekta će vam biti predstavljana iscrtkanim pravougaonikom. Otpustite taster, i program će iscrtati vaš objekat u novoj veličini. Kvadratići koji se nalaze na sredinama stranica zamišljenog pravougaonika oko objekta služe za jednostranu promenu dimenzija objekta – razvlačenje u jednu stranu. Procedura je ista kao i prethodno opisana, samo što će do promene doći samo po visini ili širini, zavisno od položaja odabranog kvadratića. Važno je uočiti razliku između ugaonih i stranačnih kvadratića, jer kod pomeranja stranačnih kvadratića dolazi do promene proporcija odabranog objekta prilikom uvećanja ili umanjenja.

Ako se već odabrani objekat klikneta još jednom, umesto kvadratićima, objekat će biti okružen sa osam strelica i malim krugom sa tačkom u sredini. Rad sa strelicama je isti kao i sa kvadratićima, ali njihove namena je drugačije. Strelice na uglovima (malo su zakrivljene) služe za rotaciju objekta. Kada zahvatite strelicu, pomerajte miša sve dok iscrtkani pravougaonik opisan oko objekta ne zauzme željeni položaj. U statusnoj liniji se može očitati za koji ste ugao zakrenuli objekat. Osa rotacije je označena malim krugom sa tačkom u sredini objekta. Ukoliko je to potrebno, centar možete „izvući" zahvatanjem mišem i odvlačenjem na željeno mesto. Ovako pomeren centar kod ponovnog selektovanja objekta se ne vraća na centar objekta (to je bio slučaj u prethodnoj verziji Corela). Strelice na sredinama stranica služe za iskošenje objekta. Zahvatite strelicu (uz horizontalnu ili vertikalnu stranicu pravougaonika) i pomerajte je dok ne postignete željeni efekat.

Selektovati možete jedan ili više objekata istovremeno. To radite tako što selektujete jedan objekat, a potom uz pritisnut taster **Shift** selektujete i druge objekte koje želite obrađivati. Tek nakon izbora poslednjeg objekta otpustite i taster SHIFT. Sada privremeno imate selektovane više objekata koje sve zajedno možete da transformišete na željeni način. Ako ponovo pritisnete taster SHIFT i kliknete mišem na neki od već odabranih objekata, on će biti izuzet iz ove grupe privremeno odabranih objekata. Ukoliko ne želite više da radite sa privremeno odabranim objektima, jednostavno kliknite negde u prostoru pored svih objekata i time ukinete privremenu selekciju objekata.

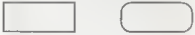
Objekte možete pomerati po radnom prostoru. Za pomeranje objekata morate biti u režimu rada sa velikom strelicom, ali nije neophodno da objekat koji pomerate bude i selektovan. Strelicu miša dovedete na željeni objekat i jednostavno uz pritisnut taster miša pomerate željeni objekat. U slučaju da objekat ima boju površine, strelicu miša postavite bilo gde na objekat, a u slučaju da objekat nema boju površine, strelicu morate postaviti na okvirnu liniju. Kod pomeranja objekata, umesto obične strelice pojavi se četvorosmerna strelica. Ona se nalazi tačno na onome mestu na kojem ste zahvatili objekat. Ako hoćete da postavite gornji desni ugao pravougaonika na tačno određeno mesto, zahvat objekta ćete otpočeti na gornjem desnom uglu, a pomeranje ćete završiti kada se presek četvorosmerne strelice nađe na mestu na koje ste želeli da postavite pravougaonik.

Ovo su bila osnovna pravila kojim ćete se služiti prilikom izbora – selektovanja objekata. Selektovanje će se praktično provlačiti kroz sva poglavlja, tako da je savladavanje rada sa ovom alatkom od bitnog značaja za opštu brzinu vašeg rada sa crtežima.

2.2 Mala strelica – Shape Tool

Mala strelica je sledeća alatka koje vam je na raspolaganju. Njen originalni naziv je **Shape Tool**, što bi u prevodu značilo alatka oblika, obrisa. Podsetimo se da je objekat sačinjen od niza čvorova povezanih linijama (bilo krivim ili ravnim). Sa ovom alatkom ćete moći da svakom pojedinačno odabranom objektu dajete sasvim novi izgled odnosno oblik, obris. Ovaj režim rada aktivirate klikom miše na malu strelicu. Potom, kao i kod selektovanja objekta, kliknite na željeni objekat i dobijate novu sliku tog objekta. Na okvirnoj liniji će vam se pojaviti novi beli, uokvireni kvadratići. Ovi kvadratići predstavljaju čvorove pomoću kojih možete raditi transformacije objekta. Vrsta transformacije zavisi od prirode objekta koji ste selektovali.

Izmena pravougaonika, zaobljavanje uglova (slika 2.1)



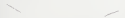
Slika 2.1

Izmena elipse, pravljenje odsečka elipse – kruga (slika 2.2)



Slika 2.2

Izmena linije/krive, menja se oblik, (slika 2.3)



Slika 2.3

Izmena teksta, menja se razmak između svih ili pojedinačnih slova (slika 2.4)

PROBA   P R O B A

Slika 2.4

Za sve vrste objekata izmene se rade tako da strelicu miša, nakon izbora objekta u ovom režimu, dovedete do čvora koji želite menjati, zahvatite ga, on se pretvara u crni kvadratić i pomerate ga tako da postignete željeni efekat.

Pošto je upotreba ove alatke vezana za poznavanje rada sa drugim alatkama, detaljan opis rada biće dat u okviru odgovarajućih poglavlja.

2.3 Zum – Zoom Tool

Ovo je jedna od pomoćnih alatki i omogućuje vam da povećate određeni deo prozora za crtanje, kako biste pojedine operacije mogli preciznije izvesti. Kada kliknete na ikonicu ove alatke, pojaviće vam se dodatni niz ikona. Značenje svake od ovih ikonica biće dato u daljem tekstu.

Ovo je prva ponuđena opcija i predstavlja uvećanje (**Zoom In**). Kada kliknete na nju, strelica miša se u prozoru za crtanje pretvara u lupu sa znakom + unutar nje. Zahvatite određeni deo vašeg crteža (zahvaćeni deo crteža će biti označen isprekidanom linijom oblika pravougaonika) i nakon otpuštanja tastera miša u radnom prostoru će vam biti prikazan samo taj deo crteža. Maksimalno uvećanje koje možete da dobijete i kvalitet prikaza zavise od karakteristika vašeg monitora. Realno maksimalno uvećanje je da jedan piksel ekrana odgovara dužini od 0.001 inča. Ova alatka se može direktno aktivirati pritiskom na taster F2.

Za razliku od prethodne alatke, ova radi umanjenje prikaza (**Zoom Out**). Izborom ove opcije prikaz će vam se vratiti za jedan korak uvećanja unazad. Maksimalno umanjenje je kada u prozoru za crtanje dobijete mogućnost prikaza stranice za crtanje od 32x32 inča. Ova alatka se direktno može aktivirati pritiskom na taster F3.

Po izboru ove alatke, program će vam dati prikaz crteža u razmeri 1:1 u odnosu prema crtežu koji će biti štampan (**Actual Size**). Tačnost ovog prikaza zavisi od tipa monitora koji imate, tako da se prvo morate uveriti da li vaš monitor daje pravu veličinu objekta izborom ove opcije.

Izborom ove alatke program će vam, iz postojećeg prikaza u okviru radnog prostora, dati prikaz svih objekata koje ste radili na vašem crtežu (**Fit in Window**). Ovo je brz način da pogledate vaš dotadašnji rad. Ova alatka se može direktno aktivirati pritiskom na taster + i F4.

Ova alatka vam daje prikaz cele stranice koja će moći da se štampa (**Show Page**). Direktno se aktivira pritiskom na tastere SHIFT i F4.

2.4 Olovka – Pencil Tool

Ova alatka omogućuje crtanje linija. Postoji više načina da crtate olovkom. Ova osnovna načina su **Freehand Mode** i **Bezier**. I za jedan i za drugi režim rada važno je znati da nacrtani objekat predstavlja otvorenu liniju (**Open Path**) koja ne može imati boju površine sve dok se ne poklope prva i poslednja tačka.

2.4.1 Režim za slobodno crtanje (Freehand Mode)

Ovaj režim rada bi se mogao prevesti kao crtanje slobodnom rukom, a omogućuje crtanje linija na dva načina. Jedan način podrazumeva da se na mesto odakle želite da crtate postavi krstić (on se pojavio u prozoru za crtanje nasuprot strelice miša nakon izbora alatke **Pencil**) koji zatim pomerate uz pritisnut taster miša. Ovakvo crtanje zahteva mnogo strpljenja i veštine. Videćete da rezultati i nisu baš sjajni, tako je ovaj način najsličniji crtanju olovkom na papiru. Drugi način je da postavite krstić na željeno mesto i kliknete mišem, pomerite krstić do mesta gde želite da završi linija, i kliknete ponovo. Ovako ste dobili liniju koja je između početne i krajnje tačke ravna. Ova linija sama za sebe predstavlja objekat. Ako želite da nacrtate izlomljenu liniju, potrebno je da ponovo kliknete u blizini završne tačke prethodne linije i da postupak ponovite. Sada će vam ova predstavljati novi objekt. Znači, potrebno je kliknuti dva puta na početku i na kraju svake linije.

Ako istovremeno držite pritisnut taster CTRL dok crtate liniju, biće vam omogućeno precizније povlačenje linija pod uglom sa korakom od 15°. Na ovaj način ćete lako crtati horizontalne, vertikalne ili linije pod određenim uglom koji je proizvod ugla od 15° (15°,30°,45° ...). Ovaj ugao se može i promeniti, o čemu će biti reči u meniju **Special**, komanda **Preferences**.

2.4.2 Režim za slobodno crtanje i mala strelica

Kada crtate liniju u *Freehand* režimu rada i nakon završetka crtanja odaberete režim rada malom strelicom, na vašoj liniji će se pojaviti čvorovi u prelomnim tačkama. Jednostavna ravna linija ima dva čvora (po jedan na početku i jedan na kraju). Čvor selektujete tako da strelicu miša dovedete na određeni čvor i kliknete. Svaki pojedini čvor moći ćete da pomerate, bilo da je on početni, krajnji ili jedan od unutrašnjih. Prilikom pomaka određenog čvora (ili više čvorova) linije između dva čvora mogu promeniti samo položaj i svoju dužinu, dok između dva čvora ostaju ravne. Ako ste crtali liniju povlačeći miša uz pritisnuti taster (kao olovkom), u prelomnim tačkama će vam se pojaviti čvorovi koji će imati i kontrolne – pomoćne čvorove. Pomoću kontrolnih čvorova možete menjati zakrivljenje linije između dva čvora. Rad sa čvorovima je opisan u odeljku 2.4.5.

2.4.3 Bezier režim

Ovaj režim rada aktivirate ako pri izboru alatke OLOVKA malo zadržite pritisnut taster miša. Pojaviće vam se dve ikone, od kojih donja predstavlja **Bezier** režim. Za razliku od prethodnog, ovaj režim podrazumeva crtanje od tačke do tačke. Postavite krstić na mesto odakle želite da počnete i pritisnite taster miša. Pomerajte dalje miša. Svaki sledeći klik označiće mesto preloma linije, ali će sve to biti jedna linija – jedan objekat. Linije između prelomnih tačaka predstavljaće ravne odsečke – linije. Ukoliko kod crtanja pojedinih tačaka linije, dok je taster miša pritisnut, izvršite i pomeranje miša, taj segment će biti nacrtan kao kriva linija. Ovo je važno kod kasnije obrade sa alatkom strelicom (**Shape Tool**). Da biste nacrtali dve razdvojene linije u ovom režimu, potrebno je da, nakon nacrtane prve linije, iz režima crtanja pređete u režim rada sa velikom strelicom i ponovo se vratite u režim crtanja. Ovo ćete najlakše uraditi skraćenom (direktnom) komandom – duplim pritiskom na taster **Space Bar**.

2.4.3.1 Bezier mode i Mala strelica

Kada crtate linije u linije zier režimu i nakon završetka

crtanja aktivirate Malu strelicu, biće vam prikazani čvorovi u prelomnim tačkama. Ukoliko ste crtali izlomljenu liniju sa ravnim odsečcima, Mala strelica vam omogućuje samo da pomerate ove čvorove. Ako ste crtali krive linije, svaki čvor će imati i svoje kontrolne – posredne čvorove. Rad sa čvorovima je opisan u odeljku 4.5.

2.4.4 Rad sa čvorovima

Pri radu sa olovkom (u **Freehand** ili **Bezier** režimu) pruža vam se maksimalna mogućnost oblikovanja linija uz pomoć čvorova. Pored onoga što je opisano u prethodnim poglavljima, imate mogućnost da aktivirate dijalog boks **Node Edit** (*pop up* meni). Njega aktivirate dvostrukim klikom na odsečak linije, između dva čvora ili na sam čvor. U zavisnosti od operacije koju želite da odradite biće vam na raspolaganju i određene komande iz ovog dijalog boksa. Pri radu sa selektovanjem čvorova, uz alatku „mala strelica" važe sva pravila za selektovanje koja su navedena i pri selektovanju sa velikom strelicom. Opšti izgled dijalog boksa dat je na slici 2.5, a u daljem tekstu će biti objašnjene njegove komande pojedinačno.



Slika 2.5

Delete Ako ste dvaput kliknuli na jedan od čvorova, ova komanda briše taj čvor. Ako ste dvaput kliknuli na liniju između dva čvora, ova komanda briše čvor koji je nacrtan poslednji sa tom odsečku linije.

Add Ako ste dvostruki klik uradili na samom čvoru, ova komanda dodaje novi čvor na sredini između tog i prethodno nacrtanog čvora. Ako ste dvostruki klik uradili negde na liniji između dva čvora, onda ova komanda dodaje novi čvor tačno na mestu klika dovde.

Break Ako dvaput kliknuli negde na liniji između dva čvora ili na samom čvoru, nakon komande Break ćete na tom mestu dobiti dva čvora koji su preklopljeni. Ako zahvatite gornji i pomerite ga u smeru prethodno nacrtanog čvora, videćete da je došlo do prekida vaše linije. Preklopljeni čvorovi su uvek tako raspoređeni da gornji pripada čvoru koji je prvi crtan. Ako ste pre izvršenja komande Break imali objekat sa zatvorenom linijom, onda je ova komanda menja u otvorenu liniju, dalje važe sva pravila za otvorenu liniju (**Open path**). Ova komanda je primenljiva i na više čvorova istovremeno.

Join Ova komanda je suprotna komandi **Break**, ona spaja razdvojene čvorove. Spajanje dva razdvojena čvora radite tako da odaberete jedan čvor, pa uz držanje tastera **Shift** odaberete i drugi čvor. Nakon toga dvostruki klik na drugi čvor uključuje meni **Node Edit**, u kojem je moguća upotreba komande **Join** (primetićete da je tada komanda **Break** isključena). Pritiskom na ovu komandu dva odabrana čvora se spajaju u jedan. Dalji rad sa ovim čvorom je kao sa ostalim čvorovima.

toLine Ova komanda je uključena ukoliko ste kliknuli na krivu liniju. Aktiviranjem ove komande krive linija na koju ste kliknuli, koja se nalazi između dva čvora, pretvara se u ravnu liniju. Izmene možete raditi na jednoj ili više krivih linija istovremeno.

toCurve Ova komanda je suprotna prethodnoj i istovremeno ne mogu biti uključene. Koja će biti uključena zavisi od vrste nacrtane linije. Ukoliko je to prava linija, moguće je da kod aktiviranja menija **Node Edit** aktivirate komanda **toCurve.** Ona će postojeću ravnu liniju pretvoriti u krivu. Dotadašnji „obični" čvorovi dobiće i po jedan ili dva pomoćna čvora (mali crni kvadrati koji su sa osnovnim čvorom vezani isprekidanom linijom). Ukoliko tada zahvatite pomoćni čvor i pomerate ga, primetićete da oblik linije – krive zavisi od položaja i udaljenosti pomoćnog čvora od osnovnog čvora. Moguće su sve kombinacije linija i krivih u jednoj izlomljenoj liniji što zavisi samo od vaših potreba. Čvorovi na spoju krive i ravne linije dobijaju samo po jedan pomoćni čvor, dok čvorovi na spoju dve krive linije dobijaju dva pomoćna čvora. Izmene možete raditi na jednoj ili više krivih linija istovremeno.

Cusp Ova komanda omogućuje da željeni broj čvorova grupišete i da ih pomerate kao jedan, u odnosu na druge koje niste odabrali u ovu grupu čvorova. Odaberite više čvorova na vašoj izlomljenoj liniji i uradite dvostruki klik na jednog od linija. Dobićete *pop up* meni sa uključenom komandom *Cusp*. Ako je aktivirate, mout se ... i tada možete, zahvatom jednog od odabranih čvorova, pomerati sve odabrane čvorove i linije koje oni obuhvataju u odnosu na ostale čvorove. Ova komanda ima uticaja samo dok je grupa čvorova selektovana. Čim ih jednom deselektujete sa ponovni **Cusp**, morate sve ovo ponoviti. Izmene možete raditi na jednoj ili više krivih linija istovremeno. Ova komanda ujedno ukida **Smooth** čvor.

Smooth Ova komanda praktično dovodi u zavisnost spojene krive i ravne linije ili više spojenih krivih linija, i čini međusobne prelaze meko zakrivljenim. Ako nacrtate izlomljenu liniju i selektujete sve njene čvorove, **Smooth** komanda će zaobliti izlomljene vrhove u zavisnosti od nagiba ravnih odsečaka. Upotrebom ove komande od testeraste linije dobićete talasastu liniju, a može se primeniti na jednom ili više čvorova istovremeno.

Align Ova komanda vam daje mogućnost da spajate čvorove objekata koji su spojeni komandom **Combine** iz menija **Arrange**. Nakon što ste nacrtane objekte kombinovali i ponovo uključili alatku „Mala strelica", biće vam prikazani čvorovi i jednog i drugog objekta. Sada izaberete čvor koji želite poravnati i uz pritisnut taster SHIFT odaberite i čvor prema kojem želite poravnanje. Nakon dvostrukog klika i izbora komande **Align** treba da odgovorite da li želite poravnanje po horizontali, vertikali ili centralnim tačkama. Ukoliko odaberete sve tri poravnanja, izabrani čvorovi će se preklopiti.

Symmet Ova komanda omogućuje da se pomoćni čvorovi postave simetrično u odnosu na osnovni čvor, ali samo kod čvorova koji spajaju dve krive linije. Komanda se aktivira duplim klikom na željeni čvor i aktiviranjem **Symmet** tastera. Nakon ove komande pomak jednog pomoćnog čvora prouzrokovaće isti takav pomak simetrično postavljenog pomoćnog čvora, sa druge strane osnovnog čvora. Izmene možete raditi na jednoj ili više krivih linija istovremeno.

Cancel Jedino ova komanda može da isključi *pop-up* meni ukoliko niste aktivirali ni jednu komandu.

2.5 Pravougaonik – Rectangle Tool

Ovu alatku uključujete tako što kliknete na odgovarajuću ikonu. Kursor se u radnom prostoru menja u krstić. Pravougaonik crtate tako što odaberete mesto za jedan od njegovih uglova, pritisnete taster miša i pomerate krstić po zamišljenoj dijagonali budućeg pravougaonika (u smeru gore-desno, gore-levo, dole-desno ili dole-levo). Sve dok držite pritisnut taster miša, možete da menjate dimenzije vašeg pravougaonika. Kao pomoć pri određivanju dimenzija pravougaonika u statusnoj liniji vam se nalaze podaci o visini i širini pravougaonika, položaju početne i trenutne (završne) tačke, kao i podatak o položaju njegovog centra. Nakon otpuštanja tastera miša u statusnoj liniji ostaje podatak da koristite alatku **Rectangle**, podaci o visini i širini, o boji površine, debljini i boji okvirne linije nacrtanog pravougaonika.

Za crtanje kvadrata procedura je ista, samo se istovremeno drži pritisnut i taster **Ctrl**. Taster miša otpustite pre tastera **Ctrl**.

Direktno aktiviranje alatke za crtanje pravougaonika postiže se pritiskom na funkcijski taster F6.

2.5.1 Alatka Rectangle i mala strelica

Ako nakon crtanja pravougaonika aktivirate malu strelicu, ova alatka će vam omogućiti da zaoblite uglove vašeg pravougaonika. Zahvatite jedan od četiri čvora pravougaonika i polako ga pomerajte od ugla prema središnu stranice. Rezultat ovog pomeranja biće zaobljeni uglovi (simetrično sva četiri). Sada ćete umesto četiri, imati osam čvorova raspoređenih po stranicama pravougaonika (svako zaobljenje završava se dva čvora). Maksimalno zaobljenje nastupa u momentu spajanja čvorova.

Ukoliko nakon izvršenog zaobljenja radite neku od transformacija objekta (širenje ili iskošenje), zaobljenje se takođe transformiše u skladu sa urađenom promenom. Ako tada niste zadovoljni rezultatom, upotrebite komandu **Undo** iz menija **Edit.**

U ovom režimu rada nemate mogućnost daljeg uticaja na oblik zakrivljenja, nije moguće aktivirati *pop-up* meni **Node Edit**.

2.6 Elipsa – Ellipse Tool

Alatku uključujete klikom na odgovarajuću ikonu. Kursor se u radnom prostoru pretvara u krstić. Crtanje je isto kao i crtanje pravougaonika. Veličina male i velike ose elipse zavisi od toga koliku elipsu ćete nacrtati, ali ose će uvek biti paralelne odgovarajućim stranama vašeg prozora za crtanje. Za crtanje kruga potrebno je da istovremeno uz pritisnut taster miša pritisnete i taster **Ctrl**. U statusnoj liniji ćete imati podatke o visini, širini, položaju početne i završne tačke i centra elipse, boji površine, boji i debljini okvirne linije.

2.6.1 Elipsa i mala strelica

Ako nakon crtanja (ili selektovanja) elipse aktivirate malu strelicu, ova alatka će vam omogućiti da nacrtate luk ili deo kruga (popularnu pitu). Deo kruga crtate tako da zahvatite čvor i pomerate ga u željenom smeru, ali tako da vam mala strelica bude unutar površine elipse. Ukoliko ovo sve ponovite, a mala strelica vam bude izvan (spolja) površine elipse, dobićete luk. U oba ova slučaja čvor elipse se pretvara u dva čvora koji među sobom zauzimaju položaj koji zavisi od položaja miša. U statusnoj liniji imaćete podatke o uglu otvorenosti elipse, o preostalom uglu do punih 360°, kao i poslednjem položaju čvora koji je pomeran. Ugao otvorenosti se izražava u stepenima, od do . Važno je znati da se svi uglovi uvek računaju u odnosu na fiksni koordinatni sistem, kod kojeg je 0° na položaju kazaljke sata u 3:00 časa, 90° na položaju 12,00 časova, 180° na položaju 9:00 časova, a 270° na položaju 6:00 časova.

Ukoliko uz zahvat i pomak čvora držite pritisnut i taster **Ctrl**, pomak ugla će odgovarati proizvodu ugla od 15°.

2.7 Alatke za kontrolu boje površine, boje i debljine okvirne linije

Ovo je značajan deo programa *CorelDraw*. Zbog dodatnog usporenja, koje je unosilo bojenje objekata, ranije verzije (koje su znatno sporije od sadašnje) nisu imale mogućnost prikazivanja boje površine i okvirne linije u prozoru za crtanje, nego samo u pregledном ekranu. Sada je to korigovano i imate mogućnost prikaza svih atributa objekta u prozoru za crtanje. Komanda **Edit Wireframe** je dodata u meni **Display**, a omogućuje prikaz u prozoru za crtanje identičan prikazu u prethodnoj verziji *Corela*.



Slika 2.6

Parametri boje površine, boje i debljine okvirne linije se pridružuju automatski novom objektu (kako je ranije definisano), ili mogu biti promenjeni tokom rada. Sami možete definisati koja će boja površine, a koja boja i debljina linije biti pridružena svakom novom objektu.

Sva slika se sastoje od objekata, a svaki objekat od okvirne linije i površine koju zatvara ta linija. Za dodeljivanje boja površine i boje okvirne linije koristi se više komandi i alatki. Crno-beli štampači će pokušati dočarati vaš izbor boje raznim kombinacijama crtica i tačkica. Nekada ćete biti pomalo i zbunjeni jer će se desiti da štampač se potpuno istim šarama odštampa dve različite boje. Ne postoji opšti savet koji bi pomogao da budu svi zadovoljni, ali se bi bilo loše da crteže koje štampate na crno-belim štampačima pravite u sivim nijansama, a slike koje ćete koristiti za prezentaciju na računaru ili za štampanje u boji pravite u željenim bojama. Broj boja i nijansi koje ćete moći videti i koristiti zavise pre svega, od kvaliteta grafičke kartice i monitora koji posedujete.

Na raspolaganju vam stoje dve osnovne grupe boja: procesne (*Process*) i gotove ili definisane (*Spot*). Gotove boje omogućavaju izbor neke od već definisanih boja. Procesne boje daju mogućnost definisanja sopstvene boje između 16 miliona boja.

Gotove boje su usklađene sa *Pantone* industrijskim standardom za identifikaciju boja, a možemo ih birati iz palete boja pri dnu radnog ekrana ili iz posebnog dijalog-boksa (slika 2.6). Ovaj dijalog boks je isti i za boju okvirne linije i za boju površine. Kod ovakvih boja možemo u dodatnom meniju definisati i zasićenje bojom.

Ako odaberete opciju rada sa procesnim bojama, bićete u mogućnosti da definišete sami svoju boju, pomoću dijalog boksova koji predstavljaju „mešalice" za boje, gde možete odmah videti izabranu boju. Na raspolaganju su vam četiri načina izrade boje, koje birate izborom komande **Model** iz dijalog boksa **Uniform Fill**, i to:

CMYK – mešanjem raznih intenziteta boja Cyan, Magenta, Yellow i Black (slika 2.7).



Slika 2.7

RGB – mešanjem crvene, zelene i plave boje (slika 2.8).



Slika 2.8

HSB – određivanjem kombinacije jačine (Hue), zasićenja (Saturation) i osvetljenosti (Brightness) boje, (slika 2.9).

Named – Ova opcija daje mogućnost izbora iz spiska postojećih boja koje ste dobili uz programski paket. U ove mogu biti uključene i boje koje vi definišete (slika 2.10).

Ovo su bile uvodne napomene koje važe i za boju okvirne linije i za boju površine objekta. Dijalog-boksevi koji su ovde prikazani nisu teški za razumevanje, a najbolje rezultate ćete postići vežbanjem. Ovaj deo nećemo ponavljati u poglavlju o boji površine objekta, jer je procedura rada ista. Neće biti reči ni o komandama za



Slika 2.9



Slika 2.10

definisanje opcija *Postscript* štampača, jer one nisu zanimljive za veći krug korisnika. Napominjemo da se tek stotine boja se definišu za *Postscript* štampače ne mogu videti iz *Corela*.

2.8 Alatka za kontrolu atributa okvirne linije

Ova alatka služi za postavljanje atributa okvirne linije. Pod atributima, koji se mogu menjati, podrazumevaju se boja i debljina linije. Ovim parametrima se mogu pridružiti unapred definisane vrednosti atributa, ili se te postavljene vrednosti mogu menjati uz pomoć palete boja ili kroz pojedine opcije iz menija ove alatke. Definisanje boje pomoću palete boja radi se na sledeći način:

- selektujete željeni objekat
- kursor miša dovedete do željene boje u paleti i kliknete desnim tasterom.

Ako kliknete na ikonicu ove alatke pojaviće vam se novi niz ikonica od kojih su neke direktne zvrške (one koje se odnose na ponuđene debljine linija i nijanse sivog), dok druge putem dodatnih menija omogućuju da sami definišete atribute.

Aktivira dijalog-boks **Outline Color**, u kojem možete definisati parametre vezane za boju okvirne linije. Ako ni je selektovan ni jedan objekat, bićete u mogućnosti da definišete boju okvirne linije koja će se automatski pridruživati svakom novom objektu. U tom slučaju, biće vam ponuđeno da ovu boju definišete za sve objekte tekstualne objekte i ostale objekte kroz dijalog-boks sa slike 2.11.



Slika 2.11

Izborom jedne opcije (najčešće je to prva) dobićete novi dijalog-boks u kojem ćete moći precizno definisati boju okvirne linije. Rad se svodi na korišćenje dijalog-boksova opisanih u poglavlju 2.7.

Aktivira dijalog-boks **Outline Pen** (slika 2.16), pomoću kojeg možete sami definisati sve atribute okvirne linije. Ako nije selektovan nijedan objekat bićete u mogućnosti da definišete parametre vezane za okvirnu liniju, debljinu i boju, na način kako je to već definisano u prethodnom pasusu.

Za ove komande postoje i direktni tasteri za aktiviranje: taster **F12** aktivira dijalog boks **Outline Pen**, dok **Shift+F12** aktivira **Outline Color**. Ove dve komande su praktično sastavak veze sa prethodnom varijom i neće ovde biti posebno i detaljno razmatran, jer se u okviru novih *roll-up* menija sadržane mogućnosti i ovih dveju komandi.

Jedna od mogućih opcija okvirne linije je da nema debljinu, odnosno da uopšte nema okvirne linije.



U okviru menija su ponuđene sledeće debljine linija 0.1, 0.7, 2.8, 5.6, 8.5 mm. Ako kliknete mišem na jednu od ponuđenih debljina, odabrani objekat će poprimiti okvirnu liniju te debljine, što možete proveriti u statusnoj liniji.



U okviru ovog menija birate boju okvirne linije. Ponuđene su boja (što je različito od izbora – bez boje), crna, i 5 gradacija sive boje (10%, 30%, 50%, 70%, 90%).

Roll-up menu – Ovo je jedna od prijatnih novina ove verzije *Corela* i treba je malo detaljnije opisati. Ukratko, u okviru ovog menija (slika 2.12) su sadržane sve komande vezane za menjanje i postavljanje atributa okvirne linije. Sve ono što je bilo do sada u okviru više menija, tu je objedinjeno. Kada se jednom aktivira ovaj *roll-up* meni ostaje vidljiv sve vreme dok ga zatvorite da ga isključite. Može se proizvoljno pomerati po radnom prostoru tako da vam ne smeta i može se minimizirati – umanjiti.



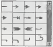


Ova linija sadrži komande identične kontrolnom meniju iz linije menija *Corela*. Sadrži i taster za umanjenje.



Slika 2.12

Debljina linije – Izbor debljine linije se vrši pomoću *scroll bar* strelica. Pritiskom (klikom) na gornju strelicu, odabirate sve deblju liniju, a pritiskom na donju strelicu odabirete sve tanju liniju. Istovremeno ćete moći da vidite odabranu liniju na malom „monitoru". Na raspolaganju su vam debljine linije od 0 (objekat nema okvirnu liniju), 1/4, 1/2, 1, 2, 4, 8, 12, 16, 20 i 24 tačaka. Vaš odabrani objekat će poprimiti željenu debljinu okvirne linije tek ako pritisnete taster Apply. Svaki sledeći odabrani objekat poprimaće istu debljinu nakon pritiska na komandu **Apply** ukoliko u međuvremenu ne menjate debljinu linije.



Slika 2.13

Oblik završetaka linija (strelica) – Ova komanda je podeljena u dva dela tako da omogućava editovanje početka i kraja linije posebno. Ovo je naročito zanimljivo kada treba da nacrtate strelice kote na crtežima, i sl. Ako kliknete na levu ikonicu koja predstavlja ujedno i prikaz odabranog završetka, prikazaće vam se slikoviti meni (slika 2.13) koji daje pripremljeni set znakova kojima može završiti vaša linija. Klikom na određeni znak moguće je promeniti postojeći. Ista procedura važi i za desnu stranu. Odabrani objekat će ove atribute poprimiti tek nakon primene tastera **Apply**. Ukoliko budete želeli neke dorade na ponuđenim znakovima, to ćete morati da uradite preko dijalog-boksa **Edit**, o čemu će biti reči malo kasnije.



Vrsta linije (Style)

Kao i prethodna komanda, i ova slikovito prikazuje prethodno postavljeni stil – vrstu linije. Klikom na ovaj taster dobijete mogućnost izbora jedne od ponuđenih linija (slika 2.14). Klikom na odabranu vrstu linije dobijete njen prikaz, a odabrani objekat poprima ovaj atribut tek nakon primene komande **Apply**.



Slika 2.14



Slika 2.15

Izbor boje linije – Boja ovog tastera odgovara onoj koja je prethodno definisana. Kada kliknete na njega, dobićete paletu boja (slika 2.15) iz koje možete da odaberete onu koju želite. Nakon klika, odmah se vraćate u početni *roll-up* meni, a taster sa bojom poprima odabranu boju. Vaš odabrani objekat će poprimiti novu boju tek nakon izvršenja komande **Apply**.

Taster **Update From...** omogućuje da za selektovani objekat kopirate parametre boje sa nekog drugog objekta na vašem crtežu. Nakon aktiviranja ove komande, biće vam prikazana strelica sa natpisom „From?" Nju dovedite do okvirne linije objekta sa kojeg želite kopirati parametre. Kliknete na njega, a efekti komande će biti vidljivi tek nakon aktiviranja komande **Apply**.



Slika 2.16

Taster **Edit** omogućuje vam potpuno kreiranje atributa linije uz pomoć **Outline Pen** dijalog-boksa (slika 2.16). Ovde se neke opcije ponavljaju, ali ima i novih. Izbor boje se radi pomoću dodatne sličice po već opisanoj proceduri. U dijalog-boksu se vidi da možete izabrati i opciju **More**. Ova opcija podrazumeva prelazak u nove dijalog-boksove koji omogućuju definisanje svih tipova boja. Ovaj deo je bio opisan u odeljku 2.7. Boja u *Corelu*, a procedura rada je identična tamo opisanoj. Opcija **Arrows** omogućuje izbor vrste završetaka linije (procedura rada je kao ranije opisana) i omogućuje određena podešavanja za svaki kraj posebno. Ova podešavanja podrazumevaju: ukidanje postojećeg završetka linije (**None**), prebacivanje završetka na drugu stranu (**Swap**), editovanje odabranog znaka (**Edit**), brisanje odabranog znaka iz postojeće liste (**Delete From List**). Komanda **Edit (Arrows)** pruža mogućnost da podesite veličinu i razna centriranja izabranog završetka – strelice. Podešavanje debljine linije (**Width**) je sasvim jasno uz postojeću sliku. Zanimljivo je to što ovde imate i mogućnost da promenite jedinicu mere (milimetar, inč, piksel, tačka odnosno *point*). Komanda **Style** je potpuno identična kao i prethodno opisana. Ovaj meni vam daje mogućnost da definišete oblik uglova (**Corners**) objekta koji sadrži izlomljenu liniju. Ovo nije ista opcija kao ona kod alatke „mala strelica", gde ste mogli da radite zaobljenost ugla pravougaonika. Zakrivljenost se odnosi na definisani oblik samog ugla i nije promenljiva. Pruža vam se mogućnost da klikom miša na određenu opciju odaberete oštre, zaobljene i odsečene uglove. **Line Caps** daje mogućnost izbora linije kod koje dužina ne zavisi od debljine linije i one kod koje debljina linije utiče na dužinu linije (zaobljeni i ravni krajevi). **Calligraphy** vam daje mogućnost da zakretanjem (**Angle**) i razvlačenjem (**Stretch**) postignete kaligrafski efekat. Na slici 2.18, dati su primeri elipse nacrtane sa **angle** 0° i **strech** 20% i elipse sa **angle** 45° i **strech** 20%.



Slika 2.18

Taster **Default** vraća promenjene vrednosti na **strech** 100% i **angle** 0°. U okviru promene atributa linije, postavljene su i ove dve naredbe: **Behind fill** i **Scale Width Image**. Kada je uključen **Behind Fill**, okvirna linija u objektu učestvuje praktično samo sa polovinom svoje širine i to spoljnjom polovinom. Ova je zanimljivo isprobati kod rada sa slovima.

Ako je uključena komanda **Scale Width Image**, debljina okvirne linije objekta će biti proporcionalno promenjena ukoliko menjate veličinu objekta. I ova komanda je vrlo korisna kod rada sa slovima, jer eliminiše neprijatne efekte nečitkosti slova kod umanjenja.

Sve navedene komande *roll up* menija vrlo lako ćete isprobati i uvežbati. Oni koji su koristili prethodne verzije *Corela*, sigurno će se malo duže navikavati na rad sa *roll-up* menijem, ali će i oni vrlo brzo prihvatiti sve prednosti ovoga rada.

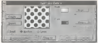
2.9 Boja površine – Fill Tool

Kada je bilo reči o objektima, rečeno je da se objekat sastoji od okvirne linije i površine koju zatvara ta linija, odnosno koje površine. Koju ćete boju površine odabrati za vaš objekat zavisi od vas i mogućnosti prikaza vašeg ekrana. Pritiskom na ikonu **Fill Tool** (kanta sa bojom) biće vam ponuđen novi meni, sličan kao i kod alatke za izbor boje i debljine okvirne linije. On sadrži sledeće opcije – komande:

Izborom ove ikonice biće vam prikazan dijalog-boks **Uniform Fill** (slika 2.6). Na raspolaganju su vam sve mogućnosti koje su opisane u odeljku 2.7.

Isto kao i kod okvirne linije, i ovde imate mogućnost da odaberete opciju „bez boje". Objekti čija je boja površine ovako određena su providni. Ukoliko i okvirnu liniju i boju površine ovako definišete, može vam se desiti da vaš objekat prividno nestane.

Izborom ove ikone program vam daje mogućnost da boju površine definišete pomoću unapred definisanih dvobojnih šara – bitmapiranih tekstura (*pattern*). Odmah dobijate i odgovarajući dijalog-boks (slika 2.19 i 2.20), pomoću kojeg možete izabrati jednu od unapred ponuđenih boja površine (tekstura) ili same kreirati svoju šaru ili teksturu. Najveći deo prostora zauzima kvadrat sa prvom od tekstura koja program nudi. Vidi se da teksturu čine dve boje: boja šare (**Front**) i boja pozadine (**Back**).



Slika 2.19



Slika 2.20

Boju svake od navedenih, možete birati i menjati pritiskom na tastere označene sa **Back** i **Front.** Ovi tasteri poprimaju boju koju ste odabrali, a ispod njih je upisan i naziv boje. Pritiskom na jedan od tastera, dobićete paletu boja. Jednostavnim klikom miša na željenu boju, ona će biti i odabrana. Pritiskom na taster **More**... dobićete dijalog-boks (slika 2.6). Ako ne želite da koristite ponuđenu teksturu, kliknite na nju i biće vam prikazan novi prozor sa svim ponuđenim teksturama (slika 2.21).



Slika 2.21

Svoj izbor završavate klikom na željenu teksturu, a program vas vraća u prethodni dijalog-boks (slika 2.19). Nakon izbora šare imate mogućnost da grubo definišete veličinu uzorka teksture (Small – mala, Medium – srednja, Large – velika). Ukoliko želite sami da odredite veličinu uzorka teksture, odaberite komandu **Tiling.** Dijalog-boks će biti proširen (slika 2.20) i moći ćete da menjate sledeće karakteristike vaše teksture.

- **Tile size** Veličinu teksture možete da definišete sa velikom tačnošću i određivati u standardno ponuđenim jedinicama mere za dužinu.
- **First Tile Offset** Položaj prvog reda tekstura po horizontali i vertikali. Znači da možete definisati da vam na ivici objekta tekstura počinje od sredine, početka, i slično. Ukoliko se radi o specifičnoj teksturi, ove može biti značajno za krajnji rezultat rada.
- **Row/Column Offset** Kolonski raster, daje mogućnost da sečete vašu teksturu po kolonama i da svaku drugu kolonu pomaknete.

Ako od ponuđenih tekstura niste uočili ni jednu koja vam odgovara, možete kreirati svoju vlastitu. To radite izborom tastera **Create..** Na ovaj način ćete ući u **Bitmap Pattern Editor**, koji vam daje mogućnost da na mreži rastera 16x16, 32x32 ili 64x64 kvadratića definišete vašu teksturu. Na raspolaganju su vam i olovke raznih debljina – 1x1, 2x2, 4x4, ili 8x8 tačaka. Nakon završenog posla, potrebno je da pritisnete samo taster OK. Vaša tekstura će automatski biti dodata na kraj reda ponuđenih tekstura (slika 2.21), i moći ćete da je menjate kao i sve prethodno ponuđene teksture.

Taster – komanda **Import**... omogućuje da unesete sa nekog od memorijskih medija već pripremljene teksture. Dijalog-boks koji će se pojaviti izborom ove komande predstavlja standardan dijalog-boks za rad sa fajlovima, jednostavan je za rad, a biće detaljno objašnjen u okviru menija **File** (slika 3.2).

Taster **Postscript options** daje dodatne mogućnosti za definisanje parametara za rad sa **Postscript** štampačima. Ovo najčešće nije interesantno za širi krug korisnika i neće biti posebno opisano.

Nakon svih podešavanja, pritiskom na taster OK, u dijalog-boksu **Two-Color Pattern** (slika 2.20) tekstura će biti primenjena na odabrani objekat, što će biti propraćeno i informacijama u statusnoj liniji.

Izborom ove ikone bićete u mogućnosti da boju površine definišete uz pomoć vektorski definisanih tekstura (**Vector Patterns**). Nakon izbora ove opcije, biće vam prikazan dijalog-boks **Full Color Pattern** (slika 2.22). Centralno mesto na njemu zauzima kvadrat sa jednom od ponuđenih vektorskih tekstura. Ako kliknete na njega, dobijate u novom prozoru sve ponuđene uzorke vektorskih tekstura. Klikom na opciju OK, odaberete željenu teksturu. Iz glavnog dijalog-boksa sada možete da odaberete malu, srednju i veliku teksturu (**Small, Medium** i **Large**) klikom na željenu veličinu. Taster **Tile** ima potpuno isto dejstvo kao i kod bitmapiranih tekstura. **Load** opcija vam daje mogućnost da uz pomoć **Load-Full Color Pattern** dijalog-boksa učitate vektorsku teksturu sa drugih medija. Rad sa ovim dijalog-boksom je isti kao i sa svim ostalim dijalog-boksovima za rad sa fajlovima i ovde neće biti posebno objašnjen. U glavnom spisku koji dobijate nalaze se one vektorske teksture koje nisu u okviru ponuđenih, a ima te mogućnost da koristite i neke teksture koje ste vi prethodno definisali preko komande **Create Pattern** iz menija **Special**. Uz pomoć tastera



Slika 2.22

Import, dobićete mogućnost da uvezete u vaš crtež vektorsku teksturu iz drugih programa u sledećim formatima: *.wmf, *.dxf, *.gif, *.cgm, *.gem, *.plt, *.pif, *.eps, *.pic, *.pct, *.tga, *.tif i *.txt. U okviru opcije bojenja objekta vektorskom teksturom, nije ponuđena izmena boja, jer su same teksture već obojene svojim bojama.

Izborom ove opcije, biće vam omogućeno da objekat obojite nijansama boja koje mogu biti postavljene linearno i radijalno uz pomoć **Fountain Fill** dijalog-boksa (slika 2.23).



Slika 2.23

Ovaj dijalog boks vam omogućuje da izaberete linearno ili radijalno raspoređivanje boja, da pomoću tastera **From** i **To** odaberete početnu i završnu boju. Taster **More**... vas uvodi u već poznate dijalog boksove za definisanje boja. Ako odaberete **Linear**, nudi vam se da linija prelaza jedne boje u drugu bude horizontalna. Ugao ove linije možete menjati uz pomoć **Angle** komande *Scroll-bar* strelicama ugao ove linije menjate za 5°. Na 3 sata se nalazi položaj od 0°, na 12 časova se nalazi položaj od 90°, na 9 časova se nalazi položaj od 180°, a na 6 časova se nalazi položaj od 90°. Položaji sa znakom „-" daće vam obrnut redosled boja. Ako strelicu miša dovedete u područje kvadrata koji vam daje primer slaganja vaših boja, on se pretvara u krstić. Ako sada pritisnete dugme miša i, držeći ga tako pritisnutog, pomerate krstić po ovom kvadratu, ugao linije prelaza će se trenutno menjati, u zavisnosti od položaja krstića, što pokazuje linija koja ide od centra kvadrata. Istovremeno ćete moći da očitate postavljeni ugao ispod komande **Angle**. Kada otpustite taster miša, moći ćete da vidite rezultat promena. U slučaju da je aktivirana opcija **Linear**, komande za **Center Offset** su isključene. Komanda **Edge Pad** omogućuje da definišete mekoću prelaza. Za vrednost 0%, mekoća odnosno širina prelaza je najveća.

Maksimalna vrednost je 45% što praktično odgovara samo jednoj uskoj liniji prelaza između početne i završne boje. To može da se predstavi i na sledeći način: za vrednost **Edge Pad** 45%, početna i završna boja zauzimaju 90% površine objekta, a predviđene boje u procesu zauzimaju samo 10% površine objekta.



Slika 2.24

Ako odaberete **Radial** opciju u kvadratu koji daje primer slaganja boja, doći će do promena (slika 2.24). Sada je prelaz od početne do završne boje postavljen radijalno, odnosno linija prelaza je krug. Kod ovakvog prelaza proces se završava tačkom koja ima boju završne boje. Početnu i završnu boju odaberete kao i kod opcije **Linear**, a komanda **Angle** je isključena. Kod opcije **Radial** možete menjati položaj centra oko kojeg će vam se realizovati proces prelaska boja. Ovo možete raditi uz pomoć *scroll-bar* strelica za vertikalni i horizontalni pomak. Pomak je izražen u procentima, tako da 50% predstavlja pomak centra završne boje iz centra kvadrata na jednu od ivica kvadrata, zavisno od predznaka promene. Kao i kod opcije **Linear**, ove promene možete da radite pomoću miša. I u ovom slučaju komanda **Edge Pad** definiše mekoću prelaza iz jedne boje u drugu.

U ovom dijalog-boksu vam je omogućeno da radite sa **Postscript** štampačima, ali sve komande, kako je već rečeno, neće biti opisane.

Ova opcija omogućava rad sa teksturama za **Postscript** štampače. Program vam dozvoljava da definišete određene parametre, ali u pregledi nom ekranu će to biti prikazano samo teksturom koja se sastoji od „PS".



Ova opcija vam daje mogućnost da direktno izaberete jednu od ponuđenih nijansi sive boje (10%, 30%, 50%, 70% i 90%), kao i belu i crnu boju površine za vaš objekat. Verovatno ćete je vrlo retko koristiti, pošto ovo isto možete uraditi i iz palete boja na dnu prozora za crtanje.

Ova opcija vam daje mogućnost da uz pomoć *roll-up* menija (slika 2.25) definišete sva gore navedena parametre za boju površine odabranog objekta. I ovaj *roll up* meni ima opšte osobine iste kao i prethodno opisani (nakon aktiviranja nalazi se stalno u radnom prostoru, dok ga vi ne postavite, ove komande **Update From**, **Edit** i **Apply**, može se minimizirati i može se po završenom poslu isključiti). Osnovne celine predstavljaju praktično ponovljeni meni za boju površine s tom razlikom da ne sadrži ponuđene vrednosti za nijanse sivog. Paletu boja koristite direktno. Ukoliko ste odabrali objekat, vaš objekat će trenutno poprimiti boju koju kliknete na ovoj paleti. Ukoliko niste odabrali ni jedan objekat, biće vam ponuđeno da definišete boje novih objekata. Komande **Update From...** i **Apply** rade potpuno isto kao i komande opisane u odeljku o boji okvirne linije iz *roll up* menija. Komanda **Edit**... vas uvodi u već poznate dijalog boksove za izbor boje (slika 2.6).



Slika 2.25



Slika 2.26

Izborom dvobojne bitmapirane teksture **Two Color Pattern** (slika 2.26) dobijate mogućnost da direktno menjate boju teksture i boju senove. Uz pomoć **Tile** komande menjate gustoću odnosno veličinu teksture, ali na drugačiju način nego što je to prethodno objašnjeno. Kada izaberete **Tile** komandu, na vašem objektu će se prikazati dva spojena kvadrata koji predstavljaju veličine dva uzorka vaše teksture. Ako strelicom miša zahvatite tačku spoja ova dva kvadrata, možete kontinuirano menjati veličinu – gustoću vaše teksture. Rezultat promene ćete moći da vidite nakon aktiviranja komande **Apply**. U okviru komande **Edit** biće vam ponuđen dijalog-boks (slika 2.6). Sve promene koje radite u ovom *roll-up* meniju, biće vidljive tek nakon komande **Apply**.





Izborom **Full Color Pattern** vektorske teksture boje površine – dobijate *roll-up* meni na slici 2.27. Rad sa njim je vrlo sličan radu sa osnovnim menijem za bojenje površine vektorskim teksturama. **Tile** komanda omogućuje podešavanje gustoće teksture. **Edit** komanda vam otvara dijalog-boks – **Full Color Pattern** (slika 2.22). Sve urađene promene su vidljive tek nakon primene komande **Apply**.

Izborom opcije **Fountain Fill** (slika 2.28), pružaju vam se mogućnosti kao i u radu sa dijalog boksom sa slike 2.23, samo sa specifičnostima vezanim za rad sa *roll-up* menijima.

Opisi ovih *roll-up* menija nisu bili detaljni, jer bi tada trebalo ponoviti puno već napisanog. Važno je znati da *roll-up* meniji uglavnom ne donose ništa novo u mogućnostima *Corela*, već samo doprinose komforu rada. Ako ste dobro savladali ono što je ranije opisano, ovo vam neće predstavljati posebnu teškoću za razumevanje.

2.10 Slovo (alatka za tekst) – Text Tool

Ovo je alatka koja vam daje mogućnost da pišete tekst u okviru vašeg crteža. Postoje dva vida izvršenja ove komande. Ako kod izbora za kratko zadržite pritisnut taster miša, na ekranu će se pojaviti dvostruka ikonica koja omogućuje izbor rada sa tekstom ili sa simbolima. Postavljanjem strelice miša na određenu ikonu i klikom miša možete izabrati jedan od ova dva režima rada.



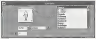
Slika 2.29

2.10.1 Alatka za ispis teksta

Ako izaberete tekstualni režim rada, imate mogućnost da tekst ispišete na dva načina: i to kao **Artistic** i kao **Paragraph text**. **Artistic text** se koristi za pisanje kraćih natpisa i rečenica, dok se **Paragraph text** koristi za ispis većeg teksta u okviru vašeg crteža. **Artistic text** se aktivira tako da nakon što izaberete **Text Tool**, postavite krstić na mesto gde želite da bude ispisan vaš tekst i kliknete. Pojavljuje se jedna uspravna crta koja označava mesto odakle će tekst biti ispisan i ujedno simbolizuje veličinu slova svojom visinom. Tekst sada slobodno unosite direktno u radnom prostoru, a na kao u dosadašnjim verzijama – u posebnom dijalog-boksu za editovanje teksta iz **Edit** menija. Nakon ispisa željenog teksta, vraćate se u **Select** režim rada samo direktnim izborom velike strelice (mišem). Nakon ovoga, vaš tekst postaje objekat koji možete transformisati kao i sve druge objekte. **Paragraph text** se aktivira tako da posle aktiviranja **Text** alatke, krstić postavite na mesto gde želite da bude jedan od uglova vašeg teksta. Potom pritisnete taster miša i uz pritisnut taster, pomerate krstić po zamišljenoj dijagonali pravougaonika. Kada odredite veličinu pravougaonika u koji ćete pisati tekst, otpustite taster miša. Biće vam prikazan pravougaonik prostor u kojem ćete moći pisati vaš tekst kao u nekom tekst-editoru. U ovaj prostor komandom **Import** iz menija **Edit** možete da unesete i tekst koji je rađen u drugom formatu kao ASCII niz karaktera. Izlazak iz režima **Paragraph text** ostvarujete klikom na veliku strelicu. Sada vam je ceo tekst koji ste uneli jedan objekat i sa njim ćete moći izvršavati sve transformacije za promenu objekta. Vrlo brzo ćete primetiti da se tekst predstavljen kao paragraf ne ponaša kao kao i objekat kojeg čini **Artistic text** režim rada. Kod uvećanja, umanjenja i sl. doći će samo do promena oblike i dimenzija okvira u kojem tekst pisan. U okviru će se tekst samo drugačije slagati. Kombinacijom promene veličine i oblika okvira paragraf teksta i tipa i veličina slova, mogu se postići vrlo zanimljivi rezultati. Sve što je vezano za promene parametara samih slova biće objašnjeno kroz rad sa dijalog-boksovima komande **Edit text** iz menija **Edit** i **Character** iz menija **Text**.

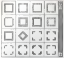
2.10.2 Alatka za rad sa simbolima

Rad sa simbolima će vam sigurno u prvom periodu pružiti mnogo zadovoljstva. Šta predstavlja rad sa simbolima? U okviru instaliranja, *Corel* je formiran zajedno sa bibliotekom sličica vezanih za meni **Symbol**. Ova alatka se koristi da iz postojeće biblioteke u vaš crtež prebacite neke od već pripremljenih slika. Prvo treba izabrati odgovarajuću ikonicu (zvezdu) alatke **Text Tool**. Nakon ovoga, krstić dovedete do mesta na kojem želite da se pojavi vaš simbol i kliknete tasterom miša. Tada će se aktivirati dijalog-boks **Symbols**, (slika 2.30) za izbor simbola i podešavanja parametara prikaza svakog pojedinog simbola. Rad sa ovim dijalog boksom se sastoji od dve celine: izbora simbola i podešavanja parametara za ispis odabranog simbola u vašem crtežu.



Slika 2.30

U dijalog-boksu je uočljiv kvadrat sa trenutno izabranim simbolom i prozor u kojem se nalazi spisak biblioteka simbola. Svaka biblioteka sastoji se od određenog broja simbola. Symbol # označava broj pod kojim se izabrani



Slika 2.31

simbol vodi u pripadajućoj biblioteci. Size daje veličinu stranice kvadrata u koji je ucrtan simbol, a može se menjati u jednoj od ponuđenih jedinica mera. Uz pomoć miša odaberete biblioteku iz koje ćete uzeti neki simbol. Tada će vam u kvadratu sa sličicom biti prikazan prvi simbol iz te biblioteke. Klikom miša na ovu sličicu otvoriće se novi prozor (slika 2.31), u kojem će umanjeno biti prikazani svi simboli koji su sadržani u toj biblioteci. Simbol odabirate klikom miša, a uz pomoć *scroll bar* tastera pretražujete i ostatak biblioteke simbola. Izabrani simbol se tada pojavljuje u kvadratu kao sličica. Izborom tastera OK, izabrani simbol će biti smešten na željeno mesto u prozoru za crtanje. Ovaj simbol postaje sada objekat (najčešće je to složeni objekat), koji se može tretirati kao i svi drugi objekti. Ovi simboli su zanimljivi jer kombinacijom pojedinih već pripremljenih sličica možete složiti vaš crtež. Ovi crteži ne gube na kvalitetu izvučenih linija i kod vrlo velikih uvećanja. Komandom **Break Appart** iz menija **Arrange** (ova komanda će biti kasnije detaljno opisana) simbol možete rastaviti na jednostavnije objekte i svaki od njih posebno obrađivati odgovarajućim komandama.

2.10.3 Mala strelica (Shape Tool) i alatka za tekst (Text Tool)

Upotreba male strelice na tekstualni objekat omogućuje podešavanje razmaka (*spacing*) između slova, reči, redova i paragrafa. Sve ove razmake možete menjati i kroz komande **Text Spacing** dijalog-boksa (opisan u 3.2.11), i to određujući tačne vrednosti za svaki pojedini razmak. Nakon uključenja ove alatke, selektovani objekat umesto crnih kvadratića dobija nove markere u obliku strelica, kojima se radi podešavanje željenog razmaka. Način podešavanja razmaka biće dat u odnosu na tekst sa slike 2.32.

2.10.3.1 Podešavanje razmaka između slova pomoću Male strelice

НОВИ РАЗМАК

Slika 2.32

Razmak između slova podešavate tako da nakon uključenja male strelice (**Shape Tool**) zahvatite marker

- u desno, ako želite da povećate razmak između slova,
- u levo, ako želite da smanjite razmak između slova.

Isprekidana linija će vam pokazivati koliko ste proširili dužinu teksta pomerajući marker. U momentu kada ste zadovoljni, otpustite taster miša i *Corel* će vam prikazati tekst sa izmenjenim razmakom između slova (slika 2.33).

Н О В И Р А З М А К

Slika 2.33

2.10.3.2 Podešavanje razmaka između reči pomoću Male strelice

Procedura rada za podešavanje razmaka između reči je ista kao i procedura rada za podešavanje razmaka između slova, s tim da istovremeno držite pritisnut i taster **Ctrl**. Rezultat je u ovom slučaju povećanje ili smanjenje razmaka između reči u vašem tekstu (slika 2.34).

Н О В И   Р А З М А К

Slika 2.34

2.10.3.3 Podešavanje razmaka između redova pomoću Male strelice

Za podešavanje razmaka između redova, zahvatite marker, nakon uključenja Male strelice, i pomerajte ovo dok vaš tekst ne dobije željeni razmak (pomak prema dole povećava, a pomak prema gore smanjuje razmak između redova) (slika 2.35).

Ако куцате текст у више редова моћи ћете да подесите размак између редова	*Ако куцате текст у више редова моћи ћете да подесите размак између редова*

Slika 2.35

2.10.3.4 Podešavanje razmaka između paragrafa pomoću Male strelice

Procedura rada je ista kao i za promenu razmaka između redova, s tim da istovremeno držite pritisnut taster **Ctrl**.

3. LINIJA MENIJA

Liniju menija čine sledeći meniji:

3.1 File menu

3.2 Edit menu

3.3 Transform menu

3.4 Effects menu

3.5 Text menu

3.6 Arrange menu

3.7 Display menu

3.8 Special menu

3.9 Help menu

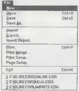
Svaki od ovih padajućih menija ima određen broj komandi, koje su aktivne ako vam je neki od objekata (ili svi) odabran. Ukoliko određena komanda u trenutku otvaranja menija nije dostupna za rad, ona će biti ispisana drugom bojom. Aktiviranje pojedinog menija ostvaruje se tako da kliknete mišem na ime odgovarajućeg menija, i nakon toga će vam se otvoriti novi prozor sa spiskom komandi koje aktivirani meni sadrži. Meniji iz ove linije se bez miša aktiviraju sa **Alt**+slovo koje je u naslovu menija podvučeno. Kursorima levo i desno idete od jednog do drugog menija. Meni sadrži i obojenu liniju (ona je obično plave boje, što zavisi od setovanja vašeg Windows-a) koju možete kursorskim tasterima – gore i dole – voditi do željene komande. Svaka komanda u menijima ima po jedno podvučeno slovo. Program vam daje mogućnost da nakon otvaranja menija komande aktivirate preko ovog slova, bez korišćenja miša.

Pored toga, određene komande možete aktivirati i kombinacijom tastera. Komande kod kojih je ovakav način aktiviranja moguć imaju u okviru menija upisanu kombinaciju ovih tastera sa desne strane komande.

3.1 Meni File

Meni **File** je prikazan na slici 3.1. Pojedine komande se aktiviraju na način koji je opisan u uvodnom delu. Ovaj meni sadrži uglavnom komande za rad sa crtežom kao fajlom.

New	**3.1.1**
Open	**3.1.2**
Save	**3.1.3**
Save All	**3.1.4**
Import	**3.1.5**
Export	**3.1.6**
Insert Object	**3.1.7**
Print	**3.1.8**
Print Merge	**3.1.10**
Print Setup	**3.1.11**
Page Setup	**3.1.12**
Exit	**3.1.12**
1., 2., 3., 4.	



Slika 3.1

3.1.1 Komanda New

Omogućuje vam da počnete rad ispočetka. Briše postojeći crtež u prozoru za crtanje, a ukoliko posle poslednjih izmena niste snimili vaš crtež, biće vam ponuđeno i da ga snimite (**Yes**) ili ne (**No**), ili odustanete od ove komande (**Cancel**). Ako se odlučite za snimanje, a crtež nema ime, biće vam ponuđeno da mu kroz **Save Drawing** dijalog-boks (slika 3.3) odredite ime i sve potrebne parametre. Svi crteži iz *Corel*-a dobijaju nastavak .**cdr** ukoliko se ne definiše nešto drugo o čemu će biti reči kasnije.

3.1.2 Komanda Open



Slika 3.2

Omogućuje vam da učitate sa nekog od memorijskih medija ranije snimljen program. **Open Drawing** dijalog-boks (slika 3.2) vam omogućuje da izaberete odakle ćete da učitate određeni program (**Drives**). Možete da definišete **File Type** (vrsta fajl formata), i to **CorelDraw! File (*.cdr)** i **Pattern File (*.pat)**. Ispod natpisa **File Name** nalazi se prozor sa spiskom fajlova definisanog formata na odabranom disku i direktorijumu. Direktorijum se bira dvostrukim klikom na njegov naziv. Na desnoj strani se nalazi kvadrat u kojem ćete moći da vidite grubi prikaz odabranog fajla. Taster **Options** daje proširenje dijalog-boksa u kojem možete da radite sortiranje fajlova po imenu i datumu. Taster **Find** omogućuje pretraživanje po ključnoj reči sa opcijom da pretraživanje vršite po svim direktorijumima, bez obzira na zadati direktorijum. Taster **Preview** uključuje i isključuje mogućnost grubog pregleda u opisanom kvadratu. Taster **Mosaic** aktivira program **Mosaic**, iz kojeg možete da vršite pregled više slika iz pojedinih biblioteka istovremeno (o ovom programu će biti nešto više reči na kraju).

3.1.3 Komanda Save

Snima vaš crtež na željeni memorijski medijum. Aktivira dijalog-boks **Save Drawing** (slika 3.3). U dijalog-boksu dobijate prikaz postojećih, već snimljenih fajlova, a na vrhu vam se nudi da upišete ime vašeg fajla u for

matu edr. Sa desne strane ovog spiska je naznačeno na kojem disku i direktorijumu se nalazite. Promenu diska radite uz pomoć komande **Drivers**.



Slika 3.3

List File Type omogućuje vam da svoj crtež snimite i u **pat** (pattern) formatu. U donjem delu daje vam se mogućnost da upišete **Keywords**, ključne reči po kojima će kasnije biti moguće pretraživati vaše fajlove. **Notes** je prostor u kojem možete da napišete neka vaša zapažanja za određeni fajl: vreme nastanka, verziju, namenu i slično. Ako izaberete opciju **Selected only** biće snimljeno samo ono što je pre aktiviranja ovog menija selektovano. Ovaj dijalog-boks se prikazuje samo kada vaš rad snimate prvi put, odnosno kada on treba da dobije prvi put ime. Svako naredno snimanje će se obavljati na način kako ste ga definisali kod prvog snimanja. **Version 2.xx** vam pruža mogućnost da vaš crtež snimite na način koji će biti prepoznatljiv i ranijim verzijama *Corela*, jednostavnim aktiviranjem ovog prekidača. Ime za staru verziju fajla more biti različito od imena fajla u ovoj verziji. Komanda **Image Header** vam omogućuje da definišete kako će prilikom snimanja biti formiran fajl za prikaz vašeg crteža u **Preview** prozoru ovog dijalog-boksa. Zavisno od toga koliko detaljan prikaz želite u ovom prozoru da dobijete prilikom učitavanja fajla imate mogućnost izbora: None (bez prikaza), 1K (mono), 2K (mono), 4K (mono), 8K (color). Za što bolju rezoluciju potrebno je više prostora i ovde je najbolji prikaz za 8K.

3.1.4 Komanda Save As...

Ova komanda je vrlo slična komandi **Save**. Razlika je u tome što vam je ovde omogućeno da snimate vaš rad pod drugim imenom ili na drugom memorijskom medijumu, u situaciji kada je već dobio ime. Procedura rada je ista kao i sa prethodnom komandom kad prvi put snimate crtež.

3.1.5 Komanda Import...

Ova komanda je veoma važna jer omogućuje unos fajlova – crteža koji su rađeni i u nekim drugim formatima. **Import** dijalog boks (slika 3.4) koji vam se ovde nudi je praktično isti kao i kod komande **Open**, ali je ovde suštno bogatija opcija **List Files of Type**. Izborom određenog formata i pritiskom na taster OK počinje proces transformacije odabranog fajla u oblik koji će vam omogućiti očitavanje u *Corelu*. Ova komanda omogućuje da u postojeći crtež ubacimo neki drugi nacrtan takođe u *Corelu*.



Slika 3.4

3.1.6 Komanda Export

Ova je komanda suprotna komandi **Import**. Ona omogućuje da crtež nacrtan u *Corelu* snimite i u nekom drugom formatu, čitljivom za druge programe. Ovde je dat za rad isti dijalog boks kao i kod komande **Import**, takođe sa povećim spiskom formata kojima možemo i snositi naš crtež u druge formate samo što će rezultat rada biti snimanje crteža u novom formatu na željenom memorijskom medijumu.

3.1.7 Komanda Insert Object

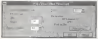


Slika 3.5

Omogućuje da u vaš crtež unesete - ubacite, crtež rađen u nekom od drugih *Windows* programa. U okviru ponuđenog **Insert Object** dijalog boksa (slika 3.5) **Object type** vam daje spisak programa iz kojih je moguće uneti objekat. Pritiskom na taster **Insert**, biće aktiviran odabrani program u njemu možete nacrtati ili iz njega preuzimiti određeni fajl. Kada budete zatvarali ovaj program biće te upitani da li želite da svoj rad prebacite u *Corel* ili ne. Ako se odlučite za prebacivanje ovaj program se zatvara, a njegov radni ekran se ubacuje u *Corel*. S obzirom na kvalitet crteža koje ćete dobijati u ponuđenim programima i kvalitet slika u *Corelu* vrlo verovatno ćete raditi ubacivanje objekata jedino iz programa *CorelPhoto-Paint* i *Microsoft Graph*.

3.1.8 Komanda Print

Ovo je komanda koja treba sav vaš uloženi trud da materijalizuje. Svi napori da nešto nacrtate, ukoliko to nešto ne bude i odštampano, neće imati ni veliki značaj. U **Print Options** dijalog-boksu (slika 3.6) možete da definišete vrlo parametara: **Selected object only** – omogućuje da se štampa samo selektovani deo crteža. **Fit To Page** – omogućuje da izbegnete eventualne probleme kada vam je crtež blizu ili izvan margina definisanih u okviru programa. **Tile** – omogućuje štampanje velikih crteža kao na crteža A4 formata. **Scale** – omogućuje da štampate vaš crtež umanjeno ili uvećano. **Fountain stripes** – definiše broj prelaza između boja, finoću prikaza nijansi boja. **Copies** – omogućuje da definišete koliko primeraka crteža želite. **Destination** – daje prikaz aktiviranog štampača, definiše usmeravanje štampanja u **File** i omogućuje aktiviranja **Print Set-up** komande (izbor i podešavanje štampača).



Slika 3.6

Komanda **Print to File** omogućuje, ukoliko ne posedujete štampač (ili štampač boljih karakteristika) da u smeru štampanje na disk ili disketu i tako dobijete **prn** datoteku koju kasnije možete da štampate na nekom štampaču. Taj "neki" štampač mora biti definisan i podešen pre štampanje **prn** fajla iz **Print Setup**. Primer: ako ste crtež štampali na disketu A: štampanje ćete zatvriti iz DOS-a sa komandom **A: copy /b XXXXX.prn prn**: direktno na odabrani štampač. Ova opcija je zanimljiva i ako neki crtež štampate više puta u većem broju primeraka jer neće svaki put biti potrebno da čekate dok *Corel* obradi crtež i pošalje na štampu. Komanda **Print Setup** je identična sa istoimenom komandom koja se dobija direktno iz menija **File**, pa će biti objašnjena kroz odeljak 3.1.10. Štampanje počinje kada pritisnete taster OK.

3.1.9 Komanda Print Merge

3
\IME\
\ADRESA\
\MESTO\

\ACA ALEKSIĆ\
\17 NOVA\
\BEOGRAD\

\BORA SORIĆ\
\23 NOVA\
\G MILANOVAC\

\STEVA STEVIĆ\
\78. NOVA\
\NIŠ\

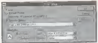
Slika 3.7

Ova komanda dodaje tekst, kreiran u programu za obradu teksa (najbolje iz *Notepad*-a) postojećem crtežu i potom sve to šalje na štampanje. Ovde nije ideja da spajate deset stranica teksta i jednu sliku, nego baš obrnuto. Ovu komandu možemo da shvatimo i kao komandu za popunu nekog obrasca, formulara, pozivnica, diploma, ili slično. Ceo posao sastoji se iz dve dele: kreiranja crteža – obrasca i kreiranja tekstualne datoteke. Crtež pravite tako da na mesta predviđena za upis teksta upisujete svoje oznake, najbolje da one budu smisleno povezane sa sadržajem koji će ih zameniti. Ako ovo prenesemo na primeru pozivnice potrebne oznake će biti IME, ADRESA, i Mesto. Ove oznake treba postaviti onako kako želite da vam bude i tekst postavljen (centriran, poravnat levo, i slično), u tip slova onaj koji želimo kao krajnji rezultat. Nakon završenog crtanja snimite vašu pozivnicu. Kada je ovo obavljeno pristupamo izradi tekst-fajla. Pređite u *Notepad* i formirajte tekst kao na slici 3.7 (važno su obrnute kose crte \).

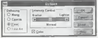
Oznake u *Corelu* i oznake u vašoj tekst-datoteci moraju da budu na isti način napisane (velika i mala slova). Broj koji na početku ove datoteke označava koliko oznaka (IME, ADRESA I MESTO) će biti korišteno. Kao što su i oznake pisane u tekst-datoteci svaka u svome redu (u stvari, može se sve pisati i u jednom redu razdvojeno sa znakom \, ali je ovako znatno preglednije i lakše za ispravljanje grešaka i izmene) tako mora biti i sa rečima koje će biti ubačene umesto oznaka. Svaka nova grupa reči (podataka) mora biti odvojena od prethodne redom razmaka. Grupa za zamenu možete imati koliko vam je potrebno. Nakon upisivanja potrebnih podataka snimite ovu datoteku i upamtite joj ime. Nakon ovoga potrebno je da učitate vašu pozivnicu i da aktivirate naredbu **Print Merge**. Otvoriće vam se standardni dijalog boks za učitavanje fajlova. Odaberite vašu tekst-datoteku i, nakon pritiska na taster **Merge**, biće aktivirana komanda **Print** (rad sa ovom komandom je opisan u odeljku 3.1.8.). Nakon izbora opcije iz ovog dijalog boksa i pritiska tastera OK, biće pokrenuto štampanje pozivnica sa svo izvanoe, čije ste podatke uneli u vašu tekst-datoteku.

3.1.10 Komanda Print Setup

Ova komanda, pomoću opcije ponuđenih u okviru dijalog boksova, pomaže da odaberete štampač i uspostavite pravilnu vezu između vašeg programa i štampača. Još kod instalacije *Windows* programa, bilo vam je ponuđeno da se odlučite koje drajvere za štampače da aktivirate tako da sada već imate neki spisak štampača. Nije loše imati drajvere za sve vrste štampača (neka se nađe) ali je nepotrebno. Nakon aktiviranja ove komande, biće vam ponuđen **Print Setup** dijalog boks (slika 3.8), podeljen na više celine. Jedna celina je izbor samog štampača. Možete birati između **Default printer** i **Specific printer**. **Specific printer** omogućuje da iz liste (formirane pri instalaciji) odaberete neki od štampača. Klikom miša na strelicu, otvoriće se novi prozor sa izborom štampača. Željeni štampač odabirate klikom miša. Prozor se zatvara i u liniji ispod **Specific printer** ostaje naziv izabranog štampača. Druga celina je **Orientation**, u odnosu na uo položaj papira pri štampanju – uspravan ili položen (**Portrait** i **Landscape**). Treća celina je **Paper** – za podešavanje komandi za papir štampača prema formatu i vrsti papira, definisanog za crtežu kroz naredbu **Page Setup** (vidi odeljak 3.1.11). Klikom na strelicu otvoriće se novi prozor sa spiskom postojećih formata papira, naravno za odabrani štampač. Izborom jednog od njih, vaš štampač će podrazumevati da treba da radi sa papirom takvog formata. **Source** definiše način uvođenja papira. Četvrta celina **Options ...** – predstavlja dopunu opcija za podešavanje vašeg štampača. Pritiskom na ovaj taster dobićete prikaz dodatnog dijalog boksa **Options** (slika 3.9) koji ne mora biti potpuno isti za sve vrste štampača. Možete podešavati kvalitet i intenzitet štampe. Ponuđene su vam četiri opcije kvaliteta štampe. Ukoliko izaberete **None**, vaš crtež će biti štampan crno-belo bez nijansi sivog. **Coarse** uključuje najbolju upis odabranog štampača. **Fine** je neka vrsta polukvalitetnog ali još uvek vrlo dobrog upisa. **Line Art** daje dobar kvalitet ali se za nju ne mogu opredijivati kod štampanja skeniranih slika. Podešavanjem intenziteta štampe podešavate nivo crnog za vaš štampač. O tome koju kombinaciju da izaberete kod određenog dokumenta biće najbolje da sami odredite i te probama. Taster **About** daje informacije o samom drajveru za vaš štampač. Pored ovih opcija, mogu se pojaviti i neke druge komande za štampača (**color**, **print quality**... i dr.) što zavisi od samog tipa štampača i njegovog drajvera. Nakon što ste odredili sve ova parametra prema vašem crtežu, sa OK zatvarate dijalog boks i možete preći na štampu.



Slika 3.8



Slika 3.9

3.1.11 Komanda Page Setup



Slika 3.10

Ova komanda omogućuje da podesite parametre izgleda i formata vaše stranice. Nakon uključenja ove komande, biće vam ponuđen **Page Setup** dijalog-boks, (slika 3.10) **Orientation** vam omogućava da izaberete uspravan ili položen (**Portrait** ili **Landscape**) format papira. Sa **Page Size** možete da izaberete jedan od ponuđenih standardnih formata papira, ili **Custom** gde vam je ponuđeno da sami definišete format papira na kojem ćete raditi. Svaki od izabranih formata propraćen je i brojnim veličinama, koje mogu biti izražene u raznim jedinicama mere. **Paper Color** definiše, pomoću već opisanih dijalog-boksova (slika 2.6) boju celog papira po kojem ćete dalje crtati. Ovde budite obazrivi kod štampanja sa jednobojnim štampačima da se nijansa papira ne poklopi sa bojama na crtežu. **Add Page Frame** crta senčen pravougaonik dimenzije odabranog formata papira. Ovaj pravougaonik će biti obojen bojom površine i bojom i tipom okvirne linije kako je definisano za nove objekte. Ovo je jednostavan način da ofarbate celu vašu stranicu određenom bojom ili bitmapiranom ili vektorskom teksturom. Tip boje ili teksture podleže definisanju kao i za sve objekte. Nakon izbora željenih opcija pritiskom tastera OK ovaj dijalog boks se zatvara i dobijate prikaz vaše stranice sa novim karakteristikama. Važno je da znate da ove promene gotovo redovno traže i promenu podešavanja odabranog štampača. Ukoliko ne postoji potrebna saglasnost ovih parametara *Corel* neće hteti da štampa vaše crteže.

3.1.12 Komanda Exit

Ova komanda omogućuje da izađete iz *Corela*. Ukoliko ste pre izlaska iz programa imali izmene na vašem crtežu a niste ih snimili, biće vam ponuđeno da ih snimite (**Yes**), ne snimite (**No**) ili se vratite ponovo u crtež (**Cancel**). Ova komanda je važna i zato što nakon napuštanja *Corela* program automatski pamti izmene parametara programa koje ste u toku rada izvršili. To su sledeći parametri:

- Parametri za podrazumevajuće boje površine objekta, debljine i boje okvirne linije.
- Parametri veličine i položaja papira.
- Parametri pomoćne mreže, koordinatni početak.
- Da li su lenjiri uključeni ili isključeni.
- Da li je uključena **Snap To Guidelines**.
- Da li je uključena paleta boja za dalji proces za crtanje.
- Da li je uključen prikaz bitmapiranih crteža ili ne.
- Da li je uključena statusna linija.
- Sva podešavanja koja se rade iz menija **Preferences**.
- Izbor jedinice mere za pojedine komande (**Page Setup, Move, Grid Frequency, Outline Pen**, i podešavanja iz menija **Preferences**.
- Koji su diskovi i direktorijumi bili poslednji izabrani u komandama. **Open, Save, Export, Import** i **Print To File**

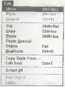
Sve ove parametre program automatski snima u CDCONFIG.SYS tako da ih dobijete ponovo kod sledećeg aktiviranja programa. Ova komanda se može aktivirati i dvostrukim klikom na ikonicu kontrol menija u gornjem levom uglu ekrana.

3.1.13 Komanda 1., 2., 3., 4.

Ovaj deo menija pamti poslednja četiri aktivirana fajla koje ste koristili u radu. Ako kod narednog aktiviranja *Corela* želite jedan od fajlova sa tog spiska, nećete morati aktivirati komandu **Open** i odgovarajuće dijalog boksove nego samo kliknuti mišem na željeni fajl.

3.2 Meni Edit

Komande iz ovog menija ćete gotovo sigurno najčešće koristiti u radu sa ovim programom. Meni **Edit** sadrži komande prikazane slikom 3.11



Slika 3.11

3.2.1 Komanda Undo

Ova komanda poništava dejstvo poslednje izvršene komande, ali samo jedan korak unazad. Koristi se najčešće kada niste zadovoljni rezultatom određene transformacije. Direktno aktiviranje ove komande je sa ALT+BAKSPACE.

3.2.2 Komanda Redo

Ova komanda je suprotna komandi **Undo**. Odmah nakon upotrebe **Undo** komanda postaje neaktivna, a aktivna postaje komanda **Redo**. Ova komanda vraća stanje koje je bilo pre upotrebe komande **Undo**. Direktno aktiviranje je sa **Alt+Return (Enter)**

3.2.3 Komanda Repeat

Ova komanda omogućuje da poslednju komandu ponovite proizvoljan broj puta sa odabranim objektom. Ako jedan objekat iskosite za određen ugao pa odaberete drugi objekat, upotrebom ove komande ćete i njega iskositi za isti ugao. Međutim **Repeat** ne ponavlja sve komande nego samo one koje su vezane za pomake, kopiranje, duplirанje, sečenje i sl. Direktno aktiviranje ove komande je sa **Ctrl+R**.

3.2.4 Komanda Cut

Ova je jedna od moćnijih komandi koja je praktično i veza između mnogih programa u *Windows* okruženju. Njome odabrani objekat kopirate u privremenu memoriju (*Clipboard*) i uklanjate ga sa crteža. Vezana je sa komandom **Paste** kojom vraćate uklonjeni objekat na isto mesto odakle je i uklonjen (u istom programu). Na ovaj način možete deo crteža ili ceo crtež da prebacite u drugi *Windows* program. Ova memorija ostaje aktivna sve vreme dok je aktivan i *Windows* iako ste recimo zatvorili *Corel*. Objekti koji su sa **Cut** kopirani na *Clipboard* mogu se proizvoljan broj puta vraćati u crtež ili drugi program sve do ponovnog aktiviranja komande **Cut** ili **Copy**. Na ovaj način možete vrlo jednostavno vašu sliku uneti u neki program za obradu teksta (*Word For Windows*) dorađivati je iz *Paintbrush*-a i sl.

3.2.5 Komanda Copy

Ova komanda je gotovo ista kao i komanda **Cut**, sa tom razlikom da ona ne uklanja kopirani objekat iz vašeg crteža. Sve navedeno za **Cut** i **Paste** komandu važi i ovde.

3.2.6 Komanda Paste

Ovo je komanda koja se koristi zajedno sa komandama **Cut** i **Copy** a omogućuje vraćanje isečenih i kopiranih objekata iz *Clipboard*-a. **Paste** možete ponavljati proizvoljan broj puta sa jednim objektom ili grupom objekata, praktično sve dok ne upotrebite komande **Cut** i **Copy** na nove objekte.

3.2.7 Komanda Paste Special



Slika 3.12

Paste Special komanda omogućuje preko odgovarajućeg dijalog-boksa (slika 3.12) da u vaš crtež ubacujete slike ili tabelove iz drugih programa, ali i da ostvarite vezu između fajlova rađenih u jednom i drugom programu (**Link**). Sa **Paste Special** nećete moći da unosete objekte koji su rađeni u vašem crtežu, nego samo one što je rađeno u drugim programima. **Source** definiše odabrani tip transformacije. **Paste** samo ubacuje sadržaj iz *Clipboard*-a, bez stvaranja veze između polaznog i odredišnog fajla. **Paste link** ubacuje sadržaj iz *Clipboard*-a, ali i stvara vezu između polaznog i odredišnog fajla.

3.2.8 Komanda Delete

Ova komanda briše selektovani objekat. Ako vam se desi da izbrišete nešto što niste želeli, upotrebite komandu **Undo**. Direktno aktiviranje ove komande je sa tasterom **Del** (objekat koji brišete mora biti selektovan)

3.2.9 Komanda Duplicate

Ova komanda duplira selektovani objekat. Duplirani objekat je pomeren (levo – desno, gore-dole) u odnosu na originalni objekat za onoliko koliko je to definisano u okviru odeljka *3.8 Meni Special*. Sa dupliranim objektom možemo raspolagati dalje kao i sa svim ostalim objektima.

3.2.10 Komanda Copy Style From..



Slika 3.13

Ovu komandu koristimo da pridružimo određene atribute sa nekog objekta selektovanom objektu. Selektujte objekat (ili više objekata) kojem želite promeniti neke od atributa. Potom aktivirate ovu komandu i kroz **Copy Style** dijalog-boks (slika 3.13) će vam biti ponuđeno da menjate debljinu i boju okvirne linije, boju površine objekta i atribute teksta (veličinu slo-

va, tip slova, prored, razmak između slova). Možete izabrati jednu, više ili sve ove opcije. Odaberite OK i pojaviće vam se nova crna strelica na kojoj piše From. Ovu strelicu treba da dovedete do objekta sa kojeg ćete koristiti definisane atribute i kliknete na njega. Tada vaš selektovani objekat (ili objekti) poprima nove vrednosti odabranih atributa. Ova komanda može biti vrlo korisna ako želite da ujednačite pojedine atribute na više objekata.

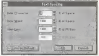
3.2.11 Komanda Edit Text

Ova komanda omogućuje da pravite izmene na teksta (**Artistic ili Paragraph**). Aktiviranje ove komande je moguće tek ako imate selektovan neki tekst objekat. U prethodnim verzijama *Corela*, ovo je bio i osnovni dijalog boks za unos teksta u tekstualni objekat. Nakon aktiviranja, pojaviće vam se **Artistic** ili **Pragraph** dijalog boks (slika 3.14) pomoću kojeg možete menjati sve parametre teksta. U najvećem prozoru ovde vidimo kompletan tekst koji je sadržan u okviru ovog tekst-objekta, bez obzira da li je on rađen u **Artistic** ili **Paragraph** režimu. U ovom prozoru ne vidite odabrani tip slova koji će biti prikazan, predviđen je za izmene samog sadržaja teksta koji ste upisali (popravka grešaka u kucanju, dodavanje sadržaja i sl.). To je praktično mali tekst-editor sa komandama koje su već standardne za gotovo sve editore teksta. U prozoru ispod komande **Style** prikazano je prvih nekoliko slova vašeg teksta, gde je vidljiv i odabrani tip slova. Tekst ispod ovog prozora označava da li su odabrana slova **True Type** ili



Slika 3.14

Wfn. Ispod komande **Fonts** nalazi se prozor koji sadrži spisak svih tipova slova koja su vam na raspolaganju. Kroz spisak se pomerate pomoću *scroll-bar* strelica, a tip slova odabirate klikom miša na željeni tip slova. Istovremeno će vam se u prozoru desno pojaviti početak vašeg teksta ispisan sa novim tipom slova. Izbor slova možete ponavljati sve dok ne dobijete željeni efekat. U delu **Justification** možete podešavati poravnanje vašeg teksta (na levo, na desno, centrirano, obostrano i bez poravnanja). Rezultat izbora biće prikazan u gornjem prozoru. Komanda **Size** vam omogućuje izbor veličine slova, sa mogućnošću definisanja u raznim jedinicama mere. Ovde imate ograničenja po pitanju najmanjih i najvećih slova. Izraženo u milimetrima, to znači od 0.3 do 508 mm. Ovo je teoretski moguće, ali verovatno nećete biti zadovoljni štampanjem slova manjih od 2 mm. Komanda **Style** omogućuje da definišete **Normal, Bold, Italic** tip slova, ukoliko odabrani tip slova ima te stilove. U donjem levom uglu nalazi se komanda **Spacing.** Izborom ove komande, dobijate **Text Spacing** dijalog boks (slika 3.15). On vam omogućuje da menjate razmak i između slova (**Inter-Character**), između reči (**Inter-Word**), između linija – prored (**Inter-Line**) i između paragrafa (**Inter-Paragraph**). Razmak između karaktera i reči dat je procentualno u odnosu na dužinu koju zauzima razmaknica. Razmak između linija i paragrafa je izražen procentualno u odnosu na veličinu slova. Ako želite da podešene vrednosti definišete kao stalne vrednosti pri svakom aktiviranju *Corela*, odaberite **Save as Default**. Ukoliko ove parametre definišete jednokratno, izaberite OK i bićete vraćeni u prethodni meni. U dijalog boksu (slika 3.14), taster **Import** aktivira već objašnjeni **Import** dijalog boks iz menija **File**. Komanda **Paste** predstavlja već poznatu komandu i pomoću nje možete da unesete tekst koji je prethodno bio upisan u *Clipboard* iz nekog drugog tekst-editora iz *Windows* okruženja. Nakon podešavanja svih ovih parametara, sa OK se vraćate u vaš radni prozor, a vaš tekst, odnosno objekat, poprima odabrane atribute.



Slika 3.15

3.2.12 Komanda Select All

Omogućuje da jednom komandom selektujemo sve nacrtane objekte. Mada ima i drugih načina da selektujete sve objekte, korisna je ako imate puno objekata raštrkanih po celom prostoru za crtanje.

3.2.13 Komanda Edit Object

Omogućuje aktiviranje programa sa fajlom koji ste u vaš crtež uneli pomoću komande **Paste Special** i definisali **Links.** Ona ima ovaj naziv, ali ćete primetiti da nije aktivna sve dok nije definisana veza sa određenim fajlom i programom. U slučaju da postoji definisana veza, umesto naziva komanda biće upisana putanja dolaska do tog fajla. Ako je aktivirate, ona će aktivirati program u kojem je odabrani objekat nastao, kao i navedeni fajl.

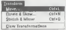
3.2.14 Komanda Links

Prikazuje listu spojenih objekata u postojećem crtežu. **Update** definiše tip veze koja je definisana u listi. **Automatic** komanda omogućuje da u *Corelu* imate uvek najsvežiju varijantu objekta iz polaznog fajla. **Manual** omogućuje izmenu u *Corelu* jedino nakon aktiviranja komande **Update Now**. **Open Source** aktivira program u kojem je rađen polazni fajl sa prikazom tog fajla. **Cancel Link** raskida postojanje veze između *Corela* i polazne aplikacije. **Change Link** preusmerava postojeću vezu *Corela* i programa polaznog fajla na željenu vezu.

3.3 Meni Transform

Ovaj meni (slika 3.16) sadrži komande za transformacije objekta. Čine ga sledeće komande:

Move	3.3.1
Rotate & Skew	3.3.2
Stretch & Mirror	3.3.3
Clear Transformations	3.3.4



Slika 3.16

3.3.1 Komanda Move

Ovom komandom vam je omogućeno da pomerate odabrane objekte. Isti efekat se postiže ako zahvatite neki objekat strelicom miša i pomerate ga po prozoru za crtanje. Razlika je u tome što ovom komandom možete tačno definisati pomak vašeg objekta. Dijalog boks vam omogućuje da definišete da li objekat pomerate u odnosu na apsolutne ili relativne koordinate. Ako je *check-box* **Absolute Coordinates** isključen, pomak se radi u relativnim koordinatama. To znači da će objekat biti pomeren za vrednost koju definišete sa **Horizontal** i **Vertical** (u ovom slučaju početne vrednosti su 0.00 za obe varijable). Ako je *check-box* **Absolute Coordinates** uključen, pomak se radi u apsolutnim koordinatama. Varijable poprimaju vrednost trenutnih koordinata objekta, i to početni položaj jednih kvadratića koji okružuju selektovani objekat. Jedan od devet kvadratića objekta možete selektovati klikom miša, a nove koordinate koje zadajete odnosiće se na tu tačku objekta. *Default* vrednost je gornji levi ugao objekta, kao najčešće korišten.



Slika 3.17

Pomak možete raditi u jednoj od već poznatih jedinica mere. Program vam daje i mogućnost da prilikom pomaka objekta original ostane na mestu uključenjem *check box*-a **Leave Original**. U gornjem desnom uglu dijalog-boksa dat vam je prikaz koordinatnog sistema sa predznacima za vrednosti prilikom pomeranja objekta. Koordinatni početak u slučaju relativnog pomaka označava mesto gornjeg levog ugla objekta, dok u slučaju apsolutnog pomaka on predstavlja koordinate od izabranog kvadratića objekta.



3.3.2 Komanda Rotate & Skew

Ova komanda omogućuje rotiranje (**Rotate**) i iskošenje (**Skew**) odabranog objekta (ili objekata). Može se realizovati zahvatanjem strelica koje se dobiju klikom na već selektovan objekat. **Rotate & Skew** dijalog boks (slika 3.18) vam pruža mogućnost da definišete ugao zakretanja objekta. Početna vrednost je 0.00° u odnosu na postojeći položaj objekta. Vrednosti sa negativnim predznakom zakreću vaš objekat u smeru kazaljke na satu, dok ga pozitivne vrednosti zakreću u smeru suprotnom od smera kazaljke sata. Možete koristiti vrednosti u rasponu između -360° i 360°. Kod iskošenja objekta možete definisati vrednosti za horizontalni i za vertikalni pomak. Pomak je posebno izražen u stepenima. I kod rotacije i kod iskošenja, možete odrediti da li želite da original ostane ili ne.

3.3.3 Komanda Stretch & Mirror

Ova komanda omogućuje razvlačenje objekta po horizontali i/ili vertikali (**Stretch**) i okretanje objekta kao u ogledalu (*Mirror*). Isti efekat se postiže ako zahvatite i pomerate strane kvadratiće na selektovanom objektu. U **Strech & Mirror** dijalog boksu (slika 3.19) vam je omogućeno da definišete razvlačenje po horizontali i vertikali izraženo u procentima.



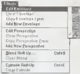
Slika 3.19

Početna vrednost je 100%, što znači da povećanje dimenzija treba definisati brojem većim od 100, dok umanjenje dimenzija traži vrednosti manje od 100. Komanda **Horz.Mirror** omogućuje da sliku objekta dobijete u zamišljenom ogledalu koje ima vertikalan položaj i prolazi kroz sredinu objekta. Komanda **Ver Mirror** daje sliku objekta u zamišljenom ogledalu, koje ima horizontalan položaj i prolazi kroz sredinu objekta. Nakon pritiska na jedan od ova dva tastera, **Stretch Horizontally** i **Stretch Vertically** poprimaju vrednosti 100 bez obzira na raniju vrednost. Ako želite pored slike u ogledalu i razvlačenje objekata, onda nakon pritiska tastera **Mirror** morate definisati i procentualno razvlačenje. I ovde program omogućuje da zadržite originalni objekat pored transformisanog. Ukoliko komandu **Stretch** radite zahvatanjem odgovarajućih kvadratića odabranog objekta i istovremeno držite pritisnut taster CTRL, razvlačenje će imati jednake efekte i na levoj i na desnoj ivici objekta (odnosno gornjoj i donjoj ivici).

3.3.4 Komanda Clear Transformations

Ova komanda omogućuje poništavanje transformacija rađenih na objektu od njegovog nastanka. Važno je da učite da ona ne ukida samo poslednju transformaciju (to se može sa **Undo** komandom) nego briše sve do tada urađene transformacije, čak i ako ste radili više koraka jedne ili nekoliko različitih transformacija. Jedino ne poništava komandu **Move**.

3.4 Meni Effect



Slika 3.20

Ovaj meni pruža mnogo mogućnosti da kreirate izgled vašeg objekta onako kako ste ga zamislili. Čine ga četiri celine (slika 3.20) i to jedna koja je vezana za rad sa envelopom ovojnicom objekta, druga vezana za rad sa perspektivom objekta, treća vezana za pretvaranje jednog u drugi objekat, i četvrta vezana za izvlačenje objekata. Svaka od ovih grupa biće objašnjena kroz pojedinačne komande.

3.4.1 Envelope

Ako se vratimo na pojam objekta i na mogućnost da sa malom strelicom (**Shape Tool**) radite izmene i detalje vašeg objekta, rad sa envelopama je samo još jedan način da postignete nove efekte menjajući oblik konturu objekta. Slično kao i kod izbora objekta velikom strelicom, envelope se prikazuje sa osam karakterističnih kvadratića (slika 3.21) koji su u obliku pravougaonika povezani isprekidanom linijom.



Slika 3.21

Zahvatanjem svakog od njih možete menjati izgled vašeg objekta. Postoje četiri načina menjanja envelope vašeg objekta. Envelopu možete editovati, brisati izmenjenu envelopu, odnosno vratiti objektu početnu envelopu, možete je kopirati sa drugog objekta, dodati novu envelopu objektu.

3.4.1.1 Komanda Edit Envelope

Izborom ova komande, envelopu objekta možete menjati na četiri ponuđena načina. Prva tri se odnose na mogućnost jednostranih izmena, dok četvrta predstavlja najveće mogućnosti editovanja envelope.



Slika 3.22

PRVI NAČIN (slika 3.22) – Izmena se radi zahvatanjem jednog od ponuđenih kvadratića i njegovim pomeranjem u željenom smeru. Primetićete da ova izmena rezultuje samo pravolinijskim deformacijama.



Slika 3.23

DRUGI NAČIN (slika 3.23) – Izmena se radi na potpuno isti način, s tim da je njihov rezultat nešto drugačiji. Primetićete da izmena envelope nije pravolinijska, kao u prethodnom slučaju već je predstavljena zakrivljenjima, zavisno od položaja strelice miša u trenutku otpuštanja tastera miša.



Slika 3.24

TREĆI NAČIN – Postupak rada je isti kao i sa prethodna načina promene envelope. Sada je rezultat dvostruka krivina na svakoj pojedinoj stranici pravougaonika izmenjene envelope. Linije na svakom kraju dobija pomoćne čvorove (kao kod editovanja sa malom strelicom) pomoću kojih možete da editujete envelopu vašeg objekta.



Slika 3.25

ČETVRTI NAČIN – Aktiviranjem ovog načina, dobijate najšire mogućnosti da menjate envelopu. U ovom slučaju karakteristične kvadratiće treba shvatiti kao čvorišta koja imaju i svoje pomoćne čvorove. Način rada je adekvatan radu sa čvorovima, koji je opisan ranije.

3.4.1.2 Komanda Clear Envelope

Ova komanda poništava urađene izmene na envelopi objekta i vraća mu originalni oblik. Ako na objektu radite sa više envelopa, onda ova komanda ima uticaja na poslednju envelopu. Ako posle editovanja envelope primenite i komandu **Perspective** (koja će biti kasnije objašnjena), ova komanda je moguća tek kada uklonite perspektivu objektu. U takvim slučajevima koristite se prvo komandom za dupliranje objekta, a potom uklonite napravljenu perspektivu. Tek tada možete aktivirati komandu **Clear Envelope**. Na tako dobijeni rad možete vratiti perspektivu sa **Copy Perspective From**, sa duplirane objekta.

Sve envelope i izmene envelope možete poništiti komandom **Clear Transformation** iz menija **Transform**.

3.4.1.3 Komanda Copy Envelope

Ova komanda kopira oblik sa jednog na drugi, željeni objekat. Objekat koji želite promeniti mora biti selektovan, a nakon aktiviranja ove komande će vam biti prikazana strelica sa pitanjem **From?** Ovu strelicu morate da dovedete na objekat sa kojeg želite da kopirate oblik i da kliknete nakon čega vaš objekat dobija oblik koji ima i odabrani objekat.

3.4.1.4 Komanda Add New Envelope

Ova komanda vam omogućuje dodavanje nove envelope objektu. Ako objektu dodajete nova envelopu podrazumeva se da je postojeći oblik (kojeg ste možda dobili izmenama originalne envelope) početni oblik i njega možete editovati na jedan od opisanih načina u prethodnim tačkama.

3.4.2 Perspective

Perspektiva je još jedna vrlo zanimljiva transformacija koja vam pruža mnogo mogućnosti izmene vašeg objekta. Ona daje utisak dubine objekta, tačnije pomoću nje ostvarujete utisak treće dimenzije vašeg objekta. Corel vam daje mogućnost definisanja perspektive sa jednom ili dve tačke.



Slika 3.26

Slično kao i kod envelope, aktiviranjem rada sa perspektivom vaš objekat će biti oivičen pravougaonikom od isprekidane linije i četiri karakteristična kvadratića, koji su raspoređeni po uglovima ovog pravougaonika (slika 3.26). Kod rada sa perspektivom potreban je naglasiti postojanje dveju karakterističnih tačaka – takozvanih tačaka nestajanja (*vanishing points*). Jedna se nalazi levo ili desno od objekta, a druga ispod ili iznad objekta i podatak o njihovom položaju ćete moći pročitati u statusnoj liniji. Jednostavno rečeno ova tačke predstavljaju mesto gde bi se spojile suprotne stranice pravougaonika koji opisuje vaš predmet. Ove tačke se mogu shvatiti i kao mesta sa kojih se posmatra vaš objekat. Početni položaj definiše da su ove tačke u beskonačnosti i ne vide se na ekranu sve dok ne promenite perspektivu objekta.

3.4.2.1 Komanda Edit Perspective

Omogućuje vam da vršite promene na objektu dodajući mu perspektivu. Promena perspektive se radi zahvatanjem jednog od karakterističnih kvadratića i njegovim pomeranjem. Zavisno koji kvadratić pomerate, birate i koju ćete tačku nestajanja pomerati. Za promenu perspektive jednom tačkom pomerajte karakteristične kvadratiće ili samo levo-desno ili samo gore-dole. Efekat promene perspektive sa dve tačke postiže se pomeranjem zahvaćenog karakterističnog kvadratića dijagonalno. Primer promene perspektive dat je na slici 3.27.



Slika 3.27

3.4.2.2 Komanda Clear Perspective

Ova komanda uklanja kreirane perspektive na odabranom objektu i daje mu oblik koji je imao pre aktiviranja perspektive. Ova komanda nema efekta kada se posle nje edituje envelopa objekta. Procedura rada je u tom slučaju slična kao i kod brisanja envelope. Prvo se mora ukloniti izmenjena envelopa, a potom se može ukloniti i perspektiva ovom komandom.

3.4.2.3 Komanda Copy Perspective From

Ova komanda omogućuje da na odabrani objekat prenesete perspektivu sa kojeg god objekta želite. Procedura rada je ista kao i kod svih komandi **Copy From**.

3.4.2.4 Komanda Add New Perspective

Ova komanda omogućuje da objektu koji ste već editovali sa komandom **Perspective**, dodate novu perspektivu, odnosno ova nova perspektiva praktično sve dotadašnje izmene perspektive podrazumeva pod normalan oblik objekta.

3.4.3 Blend

Ova transformacija svojim komandama omogućuje postepeni prelazak jednog u drugi objekat. Omogućeni su



Slika 3.28



Slika 3.29

prelaz svih kombinacija različitih objekata (po boji, obliku, veličini i slično)

Prelaz se definiše brojem koraka i stepenima rotacije, od polaznog do završnog objekta (slika 3.28). Kada radite sa objektima čija je boja površine definisana kao **PostScript**, ili kao bitmap ili vektorska tekstura, ova komanda će vršiti transformaciju, ali će oblici između početnog i završnog objekta imati belo boju površine. U ovoj verziji *Corela* je i ovde uveden *roll-up* meni (slika 3.29)

Ovaj taster zatvara *roll up* meni, ostavljajući samo naslovnu liniju.

Ovaj taster omogućuje prikaz i podešavanje broja koraka u transformaciji pomoću

Ovaj taster menja izgled *roll-up* menija i omogućuje vam prikaz spektra obuhvaćenih boja počevši od boje polaznog objekta.

Omogućuje da definišete ugao zakretanja početnog objekta u transformaciji do završnog objekta

Ovi tasteri omogućuju da odredite ili prikažete početni i svršni objekat u transformaciji

Ovaj taster omogućuje da definišete koji će čvor biti prvi kod transformacije početnog i završnog objekta. Rad sa svim tasterima je moguć pre transformacije.

taster omogućuje da se **blend** transformacija izvrši po liniji koju odaberete nakon aktiviranja ovog tastera. Programski je predviđeno da ova linija bude prava linija.

taster omogućuje aktiviranje transformacije nakon izabranih parametara transformacije

3.4.3.1 Kako definisati parametre komande Blend

Nacrtate polazni i završni objekat i odredite boje površine i okvirne linije svakog pojedinog objekta. Potom aktivirate **Blend roll-up** meni. Uz pomoć odgovarajućih tastera i posebnih strelica, odaberete polazni i završni objekat. Odaberete broj koraka i ugao rotacije polaznog objekta. Pritiskom na taster za boju, u roll-up meniju možete da definišete tip prelaza od boje površine polaznog do boje površine završnog objekta, tako što se odlučujete za tip prelaza i smer promene boje. Ovo su ujedno i osnovni parametri za definisanje **blend** transformacije. Nakon izbora tastera **Apply**, transformacija će biti i izvršena. Ukoliko ne želite da transformacija bude pravolinijska, možete to da promenite aktivirajući taster

Pre aktiviranja tastera **Apply**, aktivirajte ovaj taster i pojaviće se nova strelica miša kojom treba da dovedete do već nacrtane linije po kojoj želite da se izvrši **blend** transformacija i da kliknete

Dodatna opcija vam se pruža i kroz mogućnost izbora početnog čvora u transformaciji izborom tastera

Njegovim aktiviranjem biće vam prikazana ova strelica miša i na jednom od objekata čvorovi za editovanje. Postavljanjem strelice na željeni čvor i klikom definišete početni čvor tog objekta. Odmah će biti prikazani i čvorovi drugog objekta, na kojem isto možete definisati jedan od čvorova kao početni. Tek nakon ovoga, aktivirate komandu **Apply**. Najbolje rezultate ćete ovde postići vežbanjem. Različite kombinacije izabranih čvorova daće i različite krajnje rezultate u **blend** transformaciji.

3.4.3.2 Komanda Clear Blend

Ova komanda poništava efekat **blend** transformacije tako što briše sve oblike nastale kao rezultat transformacije i ostavlja polazni i završni objekat

3.4.4 Extrude

Ova opcija *Corela* daje utisak dubine objekta, i definiše se parametrima njegovog izvlačenja. Ova verzija *Corela*, za razliku od prethodnih, za ovu transformaciju koristi *roll up* meni koji se aktivira iz menija **Effects** komandom **Blend Roll up**. Rad sa *roll up* menijem (slika 3.30), je isti kao i sa ranije opisanim



Slika 3.30

taster je uključen nakon aktiviranja komande Extrude Roll up menu. Njegovim aktiviranjem se definišu parametri vezani za dubinu izvlačenja objekta.

Kod izvlačenja treba definisati tačku nestajanja, koja ima koordinate kao i sam papir za crtanje na kojem radite. Položaj ove tačke se vidi iz vrednosti brojeva H i V i izražen je u jedinici mere koja važi i za lenjire. Za početak rada sa ovom komandom i tačkom nestajanja, dobro je da radite sa zumom uključenim na prikaz cele strane. Pomeranje ove tačke se može izvesti i definisanjem brojeva H i V. Promena će biti vidljiva tek kada aktivirate komandu **Apply** kada ćete moći videti i samu tačku nestajanja u obliku slova x. Ako aktivirate komandu **Edit** moći ćete da vidite položaj ove tačke i pre transformacije izvlačenja. Njen položaj tada možete menjati zahvatanjem ove tačke i postavljanjem u odgovarajući položaj. Tačka nestajanja je u slučaju paralelne perspektive (**Perspective** isključeno) centrirana u odnosu na završni objekat i nalazi se u njegovoj ravni ukoliko je **To Front** uključeno a centrirana u odnosu na polazni objekat i nalazi se u njegovoj ravni kada je **To Front** isključeno. Komanda **Perspective** definiše da li će izvlačenje biti paralelno ili ne. Kada radite paralelno izvlačenje, završni objekat će biti iste veličine kao i početni. Ukoliko izvlačenje nije paralelno a komanda **Perspective** je uključena, veličina početnog i završnog objekta neće biti iste, a transformacija će odavati utisak da se sve tačke polaznog objekta spajaju ili izlaze iz tačke nestajanja.



Slika 3.31

Komanda **Depth** definiše procenat izvlačenja. Može da ima vrednosti od 1 do 99 i od -1 do -99. Predznak definiše da li će izvlačenje kod perspektive biti od i ( ) ili ka nestajućoj tački (+) u odnosu na polazni objekat

Taster uključuje novi oblik *roll-up* menija koji vam omogućuje da definišete prostornu rotaciju vašeg objekta. ??? Aktiviranjem ovog tastera, imate mogućnost promene izgleda ovog *roll up* menija u jedan od dva prikazana na slici 3.31. Jedan način rada je da nestajuću tačku pomerate u smerovima strelica koje su vam ponuđene. Drugi način daje mogućnost da definišete tačne koordinate nestajuće tačke kao i smer i ugao rotacije. Ova komanda ima dva načina delovanja – kao **Perspective Extrusion** i kao **Parallel Extrusion.**

Način upotrebe i krajnji efekti jedne ili druge rotacije zavise od tipa izvlačenja. Rad sa ovakvim menijem je sasvim jednostavan. Izborom i aktiviranjem jedne od strelica (ili za one sa većim iskustvom – upisivanjem vrednosti za pojedine smerove rotacije) jednostavno možete da rotirate vaš objekat



Slika 3.35

Aktiviranjem ove komande bićete u mogućnosti da definišete položaj izvora svetla (**Shading Effects**) u odnosu na objekat koji ste dobili izvlačenjem i eventualno rotacijom. Biće vam ponuđen novi *roll-up* meni kao na slici 3.32. Nakon uključenja prikazanog prekidača (klikom na **On**) možete definisati položaj izvora svetla koji će obasjavati vaš objekat. U ovom *roll-up* meniju možete takođe da definišete i intenzitet svetlosnog izvora koji obasjava vaš objekat. Predloženа vrednost je 100. Svetlosni izvor može da ima intenzitet od 0 do 200. Vrednost intenziteta se može definisati brojčano ili preko klizača prikazanog potenciometra.



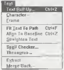
Slika 3.33

Aktiviranjem ovog tastera, otvarićete ponovo novi *roll up* meni (slika 3.33) za editovanje izvučenih objekata. Uz pomoć ovog *roll-up* menija moći ćete da definišete boje objekta transformisanog izvlačenjem. Možete definisati da li koristite boju koju ima sam objekat (**Use Object Fill**) ili boju koju sami odaberete (**Solid Fill**) iz ponuđene palete boja, ili pak hoćete da se boje senke kreću između dveju odabranih (**Shade From – To**) takođe iz ponuđene palete boja.

3.5 Meni Text

Ovo je meni (slika 3.34) koji vam omogućuje da definišete parametre vezane za rad sa tekstualnim objektima. U okviru menija **Edit** bila je reč o delu komandi za izmene teksta. Ovde je na jednom mestu koncentrisan čitav niz mogućnosti za rad sa tekstovima. U okviru ovog menija nalazi se i *roll up* meni koji pokriva najveći broj mogućnosti rada sa tekstom, kao i dodatne opcije za oblikovanje i način ispisa teksta.

Text Roll up	3.5.1
Character	3.5.2
Frame	3.5.3
Fit Text To Path	3.5.4
Align To Baseline	3.5.5
Straighten Text	3.5.6
Spell Checker	3.5.7
Thesaurus	3.5.8
Extract i Merge Back	3.5.9



Slika 3.34

3.5.1 Komanda Text Roll-up...



Slika 3.35

Već je ranije rečeno da je roll-up meni novina ove verzije Corela, a ovaj roll up meni još jednom dokazuje opravdanje njihovog postojanja. Na slici je prikazan izgled ovog menija. Svaka od njegovih komandi biće posebno opisana. Način rada sa svim roll up menijem je isti kao i sa ove roll-up menija, sa posebnostima koje su vezane za ovu konkretnu funkciju – definisanje atributa teksta. Na vrhu je naslovna linija, koja ostaje stalno vidljiva i kada je meni minimiziran. Ako strelicu miša dovedete na ovu liniju i pomerate je sa pritisnutim dugmetom miša moći ćete da pozicionirate roll up meni tako da vam ne smeta u radu sa odabranim objektom. Komande iz roll up menija imaće efekta na odabrani objekat samo ako nakon podešavanja pojedinih parametara upotrebite i komandu **Apply**.



Niz prekidača nalazi se odmah ispod naslovne linije i omogućuje vam izbor načina poravnavanja teksta i to poravnavanje uz levu ili desnu marginu, centrirano puno poravnavanje, i bez poravnavanja. Poravnavanje će imati različite efekte zavisno od toga da li selektovani objekat predstavlja **Artistic** ili **Paragraph text**. U slučaju da obrađujete **Artistic** tekst, program razlikuje dva slučaja poravnavanja:

- Ako je tekst samo u jednom redu, biće poravnat u odnosu na mesto na koje ste postavili kursor nakon izbora alatke za tekst (**Text Tool**) i kliknuli mišem. Poravnavanje uz levu ivicu u ovom slučaju znači da će početak teksta biti postavljen desno od tog mesta. Poravnavanje uz desnu ivicu znači da će se desna ivica teksta postaviti tako da stoji odmah uz to mesto sa leve strane. Centrirano poravnanje znači da će se sredina ispisanog reda postaviti na to mesto. U ovoj opciji puno poravnavanje nije moguće aktivirati, a opcija bez poravnavanja neće imati efekta.
- Ako ste tekst kucali u više redova, poravnavanje će biti realizovano na isti način kao i u prethodnom slučaju, s tim da će i svaki od unesenih redova imati odabrano poravnavanje. Kao i u prethodnom slučaju, opcija punog poravnavanja nije moguća, a opcija bez poravnavanja nema efekta na uneti tekst.

U slučaju da ste tekst uneli u **Paragraph** režimu, opcija poravnanja će imati efekat kao i u svim programima za obradu teksta, s tim da će poravnavanje biti rađeno u odnosu na ivice odabranog okvira za paragraf tekst.



tipa slova možete izabrati uz pomoć scroll bar menija.



veličina slova može biti definisana u jednoj od ponuđenih jedinica mere.



Pomoću ovog scroll bar menija možete definisati veličinu vaših slova u odabranoj jedinici mere.



Ovaj prekidač vam omogućuje da definišete **Bold** stil teksta.



Ovaj prekidač vam omogućuje da definišete **Italic** (kurziv) stil teksta.



Ovi prekidači vam omogućuju da definišete indeksni ispis vašeg teksta (**subscript** i **superscript**).



Ovo je nova komanda u Corelu. Koristi se za definisanje međuslovnog razmaka – položaja pojedinih slova ili grupe slova u odnosu na druga slova u tekstu. Primenljiva je i na **Artistic** i **Paragraph** tekst. Aktiviranje je moguće ako selektujete tekst uz pomoć alatke **Shape** i ako selektujete jedan ili više čvorova koji pripadaju slovima koja želite pomerati. Pritiskom na ovaj taster, aktiviraće se **Character Spacing** dijalog boks (slika 3.36). U okviru ovog dijalog-boksa, možete da definišete horizontalni i vertikalni pomak i ugao zakretanja odabranog ili odabranih slova.



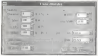
Slika 3.36



Ovo je takođe nova komanda u Corelu. Njenim aktiviranjem aktivira se i **Frame Attributes** dijalog boks (slika 3.37). Pomoću njega sada možete, kao i u okviru komande **Edit text** iz menija **Edit**, da definišete **Spacing**. Možete da definišete poravnavanje (**Justification**) i jedinicu mere (**Units**) za parametre koji se u okviru ovog dijalog-boksa definišu. Ono što predstavlja novinu kod Corela je mogućnost da sada oblikujete tekst u kolone (**Columns**). U okviru ove opcije možete da definišete u koliko kolona će vaš tekst biti ispisan (**Number**) i koliki će razmak biti između definisanih kolona (**Gutter Width**). Prijatna novina svakako je i mogućnost da definišete automatsko rastavljanje reči na kraju reda aktiviranjem opcije **Automatic Hyphenation Hot Zone** definiše udaljenost kraja linije od desne margine na kojoj će Corel uključiti rastavljanje prve reči iz sledećeg reda. Što je ova vrednost manja, poravnavanje će biti bolje.



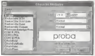
Kao i kod svih roll-up menija, rezultati promena pojedinih ili svih parametara biće vidljivi tek nakon primene ove komande.



Slika 3.37

3.5.2 Komanda Character

Komanda **Character** je još jedna od komandi koje su novina u okviru ovog menija. Iako nije sasvim jasno čemu služi pored već postojećeg **Edit text** iz menija **Edit** i roll-up menija.



Slika 3.38

Aktiviranjem ove komande dobijate sasvim nov dijalog-boks (sa starim opcijama za promenu parametara teksta) – **Character Attributes** (slika 3.38). Pošto je o svim ovim opcijama već bilo reči, nema potrebe da ih ponovo objašnjavamo. Jedino se može reći da se iza komande **Placement** skriva izbor načina ispisa teksta – **Normal, Superscript** ili **Subscript**.

3.5.3 Komanda Frame. .

Ova komanda aktivira dijalog boks **Frame Attributes** (slika 3.37).

3.5.4 Komanda Fit Text To Path



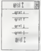
Ova komanda je postojala i u prethodnim verzijama, ali u okviru drugog menija i tada nije bila u verziji sa roll up menijem. Omogućuje vam da postavite tekst na liniju koju sami definišete. Ta linija može biti obična linija koju vi nacrtate, elipsa, pravougaonik ili bilo koja linija koju definišete. Aktiviranjem ove komande aktiviraće se roll-up meni (slika 3.39), koji vam slikovito daje mogućnost definisanja parametara. Uz već poznati način rada sa roll up menijem, evo objašnjenja pojedinih parametara.

Slika 3.39



Slika 3.41

Prvi parametar koji vam se nudi kada postavljate tekst na odabranu liniju vidi se na slici 3.40. Kao što se vidi, on definiše način postavljanja slova u odnosu na zakrivljenost izabrane linije. Ostaje vam samo da se opredelite za željeni način.



Slika 3.42

Ovaj parametar (slika 3.41) omogućuje podešavanje položaja teksta u odnosu na definisanu liniju, tačnije njegovu udaljenost od definisane linije. Pruža vam se mogućnost da tekst postavite na liniju, ispod linije, neposredno iznad linije, da tekst bude centriran sa linijom te da sami definišete položaj teksta u odnosu na liniju. Svi osim poslednjeg slučaja su jednostavni za upotrebu i njihova primena se svodi na to da program odabranu opciju izvršava nakon primene komande **Apply**. Opcija koja vam daje mogućnost da sami definišete svu udaljenost se koristi na sledeći način:

- nakon izbora ove komande od dva selektovana objekta (teksta i linije) selektovan će ostati samo tekst
- zahvatite tekst-objekat i pomerajte ga do željenog položaja, istovremeno će vam program iscrtati liniju koja simbolizuje odabranu udaljenost, bilo ispod ili iznad linije
- kada ste odabrali položaj za vaš tekst, otpustite taster miša i dobićete odmah prikaz svog rada bez pritiska komande **Apply**
- ovu radnju ćete moći ponavljati sve dok ne budete zadovoljni vašim radom, a nakon toga pritisnite taster **Apply**, čime će slaganje teksta biti završeno.



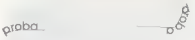
Slika 3.43

Ovaj parametar (slika 3.42) omogućuje definisanje načina centriranja teksta u odnosu na odabranu liniju. Ponuđene opcije su poravnavanje uz levi, desni i centralni deo linije. Odabrana opcija biće izvršena nakon primene komande **Apply**.



Ova komanda nakon aktiviranja tastera **Apply** menija već izvršenu podešavanja teksta u odnosu na liniju tako da tekst postavlja sa suprotne strane linije, i ispisuje ga s

desna na levo. Rezultat izvršenja ove komande vidi se na slici 3.43



Slika 3.43

3.5.5 Komanda Align To Baseline

Ova komanda ukida sve izmene koje su bile opisane u okviru **Character Kerning (Character Spacing** dijalog-boks, slika 3.36). Sve promene koje ste radili na pojedinačnim slovima ili grupama slova, ova komanda briše i vraća početni izgled vašem tekstu.

3.5.6 Komanda Straighten Text

Ova komanda ukida sve izmene rađene na tekstu — povećanje, iskošenje, rotiranje, postavljanje oko nekog objekta — i vraća tekstu oblik i položaj koji je imao pre svih urađenih izmena.

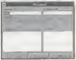
3.5.7 Komanda Spelling Checker



Slika 3.44

Ova komanda je potpuno nova u ovoj verziji *Corela* i upotpunjuje novine vezane za rad sa editorom teksta. Omogućuje proveru pravilnosti ispisa teksta, naravno za fond reči engleskog jezika (koristi se *Houghton Mifflin Dictionary* sa oko 116000 reči). Njenim aktiviranjem, pojaviće se **Spelling Checker** dijalog-boks (slika 3.44). U polje **Word to Check** treba upisati reč ili tekst koji želite da proverite, a nakon toga pritisnuti taster **Check Word** ili **Check Text**. Ako je reč dobro napisana *Corel* će vam u malom dijalog-boksu dati poruku da je reč ili tekst OK. Ukoliko reč nije prepoznata, polje **Word to Check** se menja u polje **Word not Found** a polje **Alternatives** se menja u polje **Replace With**. Ako ne želite da menjate ovu reč, izaberite komandu **Ignore** i biće vam ponuđena sledeća nepoznata reč. Ako u ovom momentu ne želite dalji rad, izaberite komandu **Ignore All**. U slučaju da želite izmenu ispisa izaberite komandu **Suggest**. Tada će vam biti omogućena da postojeću reč zamenite nekom sa ponuđenog spiska (izborom reči i komande **Replace**). Ukoliko vam nije ponuđena odgovarajuća zamena za vašu reč u polju **Replace With** upišite pravilnu reč i aktivirajte komandu **Replace**. Postoji mogućnost da vi sami definišete rečnik koji ćete koristiti prilikom rada sa *Spelling Checker*-om. Prvo postavite kurzor u polje ispod natpisa **Create** i napišite naziv vašeg ličnog rečnika. Aktiviraće se komanda **Create** koju je potrebno odabrati. U polju **Personal Dictionary** će se tada pojaviti naziv vašeg rečnika sa .dic ekstenzijom. Sada će biti moguće aktivirati i komandu **Add**. Ako upišete vašu reč u polje **Word to Check** i aktivirate komandu **Add**, *Corel* će vas obavestiti da je ova reč dodata vašem rečniku. Dijalog-boks se zatvara komandom **Cancel**, a ona neće poništiti urađene izmene, ukoliko ste ih imali.

3.5.8 Komanda Thesaurus



Slika 3.45

Ova komanda je takođe nova u *Corelu* i omogućuje zamenu postojeće reči njenim sinonimom, ali, naravno ne postoji datoteka sa podacima za naše reči. Ukoliko pišete na engleskom jeziku, upotreba ove komande je sasvim jednostavna. Rad se odvija uz **Thesaurus** dijalog-boks (slika 3.45).

U polju **Synonym for** se nalazi vaša izabrana reč za zamenu. Polje **Definition** prikazuje objašnjenje izabrane reči. Ukoliko je za odabranu reč vezano više objašnjenja, ona će biti prikazana u polju **Definitions**, a u polju **Synonyms** odgovarajući sinonimi. Izborom tastera **Replace** biće izvršena zamena izabrane reči i izabranog sinonima. Taster **Lookup** se koristi kod manuelnog unosa reči u polje **Synonym for**.

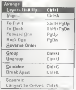
3.5.9 Komanda Extract... i Komanda Merge-Back...

Ovo su nove komande u *Corelu* koje značajno doprinose komforu rada, pogotovo ako se radi o nekom dužem tekstu. One omogućuju da tekst koji ste napisali u *Corelu* „izvučete" (**Extract**) i snimite na HDD u .txt formatu, a samim tim stvorite mogućnost editovanja tog teksta u nekom boljem tekst procesoru i da ga potom „vratite" (**Merge-Back**) u *Corel*. Da biste izvršili ove komande, potrebno je određeni tekst da selektujete. Potom aktivirate komandu, koja će vam u dijalog-boksu (poznati dijalog-boks za snimanje, slika 3.3), ponuditi da selektovani tekst snimite pod nazivom koji vi zadate u .txt formatu. Nakon ovoga možete privremeno da izađete iz *Corela* i da aktivirate neki od procesora teksta iz *Windows* okruženja (*Windows Notepad*) u kome ćete editovati vaš tekst i snimiti ga kao ASCII tekst fajl. Potom se vraćate u *Corel* i izborom komande **Merge Back** vraćate vaš editovani tekst u crtež. **Merge Back** će vam u tom slučaju ponuditi **Merge Back** dijalog-boks, koji je u stvari malo pojednostavljeni **Open File** dijalog-boks (slika 3.2).

3.6 Meni Arrange

Ovaj meni (slika 3.46) predstavlja skup komandi za definisanje odnosa između pojedinih objekata. U odnosu na prethodnu verziju *Corela* iz ovog menija je isključena komanda *Fit Text To Path* (opisana u meniju *Text*), a dodata je komanda *roll up* meni za rad sa „lejerima". Svaka od ovih komandi biće posebno prokomentarisana u narednim redovima.

Align	3.6.1
To Back,	3.6.2
To Front,	
Forward One,	
Back One,	
Reverse Order,	
Group,	
Ungroup,	
Combine,	
Break Apart	
Layers Roll-up	3.6.3



slika 3.46

3.6.1 Layers roll-up meni

Layers – slojevi su novina ove verzije *Corela*. Omogućuju vam da imate crtež u više nezavisnih slojeva, što daje više fleksibilnosti, pogotovo kod složenijih crteža. Možete definisati koliko hoćete slojeva, ali samo jedan sloj može biti aktivan za editovanje. Svaki od slojeva može da bude vidljiv ili nevidljiv. Ova opcija vam omogućuje da privremeno uklonite objekte koji bi vam mogli smetati prilikom crtanja. Pojedine ili sve slojeve možete zaključati, kako biste ih zaštitili od slučajnog editovanja, a svaki od definisanih slojeva možete štampati zasebno ako to želite.

Nakon aktiviranja ove komande iz menija **Arrange** pojaviće vam se odgovarajući *roll up* meni (slika 3.47) prikazujući tri unapred definisana sloja: **Layer 1, Guides** i **Grid. Layer 1** je inicijalni sloj koji se pridružuje svakom crtežu. **Guides** sadrži sve vodeće linije (**Guidlines**) koje koristite u vašem crtežu. Iako su svi vaši objekti smešteni u drugi sloj a ne u **Guides**, i dalje ćete moći koristiti komande za poravnavanje objekata (**Snap To Guidelines** i sl.). Ovaj sloj možete takođe da koristite za crtanje objekata, s tim da će svi objekti koji su nacrtani u njemu biti vodeći objekti. **Grid** sloj je takođe zasebna celina. On sadrži grid odnosno mrežu (u njoj nešto više u okviru menija **Display**) može se odštampati ali ne može postati aktivan za crtanje objekata – on je uvek u *Locked* režimu.



Slika 3.47



slika 3.48

Aktiviranjem ovog tastera iz *roll up* menija dobijate dodatni padajući meni sa opcijama za rad sa slojevima (slika 3.47). Komanda ovog menija za slučaju zasebno opisa.

3.6.1.1 Komanda Create New Layer

Kada aktivirate ovu komandu aktiviraće se **Layer Options** dijalog-boks (slika 3.48). On će u polju prikazati naziv prvog sledećeg sloja. Ako hoćete da definišete novi sloj na to mesto ćete upisati naziv svog novog sloja. Naziv ne sme imati više od 32 karaktera. Pritiskom na OK taster vaš novi sloj će biti prikazan u okviru osnovnog *roll-up* menija.

3.6.1.2 Komanda Editing Layers

Aktiviranje ove komande omogućuje vam da podešavate pojedine parametre sloja koji ste izabrali iz osnovnog *roll-up* menija. Ponovo će vam biti prikazan **Layer Option** dijalog-boks, u kojem na vrlo jednostavan način možete:

- da promenite ime izabranog sloja,
- definišete da li će biti vidljiv ili nevidljiv (**Visible**), definišete da li će izabrani sloj moći da se štampa (**Printable**)
- definišete da li će izabrani sloj biti zaključan za editovanje, kako biste sprečili slučajne promene već gotovih objekata (**Locked**)
- definišete boju okvirne linije koju će objekti izabranog sloja imati samo u prikazu na ekranu, dok ova boja neće imati efekte u drugim situacijama (**Color Override**). Ako je aktivirana opcija **Edit** Wireframe iz menija Dispaly objekti će uvek biti prikazani okvirnom linijom izabranom u Color Override bez obzira da li je ova opcija uključena ili ne. Ukoliko radite sa uključenom **Edit Wireframe** opcijom, objekti će biti

prikazani okvirnom linijom kao i u **Edit Wireframe** opciji samo ako je **Color Override** opcija uključena.

Dijalog-boks **Layers Options** za **Guides** i **Grid** slojeve ima dodatni taster **Setup**, koji aktivira **Guide line** i **Grid Setup** dijalog boksove koji će detaljnije biti objašnjeni kroz opis komandi **Guidelines Setup** i **Grid Setup** iz menija **Display**.

3.6.1.3 Komanda Delete Layer

Ova komanda omogućuje da izbrišete sloj koji je odabran u osnovnom *roll-up* meniju za slojeve. Istovremeno će biti izbrisani i svi objekti koje sadrži taj sloj. Brisanjem izabranog menija, odmah se aktivira sloj koji je u spisku slojeva ispod izbrisanog.

3.6.1.4 Komanda Move an Object from One Layer to Another

Ova komanda omogućuje da prebacujete objekte iz jednog sloja u drugi na sledeći način:

- prvo selektujete objekat koji želite prebaciti iz postojećeg sloja u drugi sloj;
- aktivirajte komandu **Move** iz padajućeg menija, koji se nalazi u *roll up* meniju za **Layer**;
- pojaviće vam se nova strelica sa natpisom **To?**;
- dovedite ovu strelicu na naziv sloja u koji želite prebaciti vaš objekat i kliknite.

U statusnoj liniji ćete lako moći videti da je prebacivanje izvršeno. Vaš odabrani objekat neće više biti definisan kao objekat sloja u kojem je bio.

3.6.1.5 Komanda Copy an Object from One Layer to Another

Ova komanda omogućuje da kopirate pojedinačne objekte iz jednog u drugi sloj. Procedura rada je ista kao i kod komande **Move**, samo što je rezultat drugačiji. Sada ćete na istom mestu imati odabrani objekat u oba sloja. To možete proveriti ako u jednom od slojeva pomerite vaš objekat – objekat iz drugog sloja ostaće na istom mestu.

3.6.1.6 Komanda Multi-Layering

Ova komanda omogućuje prikaz i editovanje objekata svih postojećih slojeva istovremeno. Komanda se ponaša kao prekidač – ako je aktivirate jednom, ona je uključena, a drugi put postaje isključena.

3.6.1.7 Komanda za promenu redosleda slojeva

Slojevi su u listi i prikazu na ekranu složeni po redosledu nastajanja. Ako hoćete da promenite ovaj redosled, jednostavno zahvatite mišem natpis sloja koji želite pomeriti i postavite ga na željeno mesto u redosledu slojeva.

3.6.2 Komanda Align

Ova komanda omogućuje poravnavanje objekata u odnosu na sredinu stranice (**Align to Center of Page**) u odnosu na stranicu – poravnavanje po horizontali i/ili vertikali, i poravnanje jednog u odnosu na drugi objekat.

Rad sa opcijama iz **Align** dijalog-boksa (slika 3.49) je jednostavan. Poravnavanje u odnosu na stranicu možete da radite sa jednom od ponuđenih kombinacija poravnavanja. Ukoliko želite da poravnavate jedan u odnosu na



Slika 3.49

drugi objekat, biće potrebno da selektujete objekat koji želite da poravnate, zatim objekat prema kojem želite poravnavanje, da odaberete komandu **Align** iz menija Arrange, a potom da se opredelite za jednu od ponuđenih kombinacija poravnavanja po horizontali i/ili vertikali. Poravnavanje se može raditi sa svim vrstama objekata. Nakon izbora kombinacije poravnavanja, sa tasterom OK potvrđujete svoju odluku i poravnavanje odabranih predmeta će biti izvršeno. Ovde se može reći i to da *Corel* može poravnavanja raditi i na više odabranih objekata u odnosu na samo jedan objekat.

Opcija **Align to Grid** radi poravnavanje sa mrežom (**grid**) a biće objašnjena kroz meni **Display**.

Ukoliko niste zadovoljni sa postignutim rezultatima dejstvo ove komande se ukida aktiviranjem komande **Undo**.

3.6.3 Komanda za uređenje međusobnih odnosa nacrtanih objekata

Ovaj skup naredbi (**To Back, To Front, Forward One, Back One, Reverse Order, Group, Ungroup, Combine, Break Apart, Separate, Convert To Curves**) omogućuje da uređujete međusobne položaje i odnose pojedinih objekata. Još na samom početku rečeno je da se sve slike i crteži sastoje od objekata. Zbog pojedinih osobina objekata, dešava se da neke objekte ne možete da vidite, jer su pokriveni drugim.

3.6.3.1 Komanda To Front

Ova komanda postavlja odabrani objekat na vrh ako postoji više nacrtanih objekata, bez obzira na njihov dotadašnji međusobni redosled odnosno položaj.

3.6.3.2 Komanda To Back

Ova komanda, suprotno od prethodne, odabrani objekat postavlja na „dno" odnosno ispod svih već nacrtanih objekata, bez obzira na njihov dotadašnji redosled odnosno položaj.

3.6.3.3 Komanda Back One

Ova komanda omogućuje da promenite položaj odabranog objekta i to za jedan objekat unazad. Ako je objekat bio na vrhu, sada će biti na drugom mestu.

3.6.3.4 Komanda Reverse Order

Ova komanda omogućuje da potpuno promenite redosled objekata, tako da poslednji objekat postaje prvi i obrnuto.

3.6.3.5 Komanda Group

Ova komanda omogućuje da nacrtane objekte grupišete kao jedan novi objekat. Svaka dalja transformacija ima efekat na svim objektima zajedno, sve do aktiviranja komande **Ungroup**. Komanda može biti vrlo korisna pogotovo kada radite i iz sitnih detalja koji će biti deo nekog većeg crteža. Podatke o grupisanim objektima moći ćete da pratite u statusnoj liniji. Grupisanje više objekata se obavlja tako da se istovremeno selektuju svi objekti koje želite pretvoriti u jednu grupu, a potom aktiviramo ovu komandu iz menija **Arrange**.

3.6.3.6 Komanda Ungroup

Ova komanda ukida dejstvo komande **Group** tako što odaberete grupisani objekat i aktivirate ovu komandu. „Razgrupisavanje" će biti obavljeno ali će svi do tada grupisani objekti ostati selektovani sve dok ne kliknemo mišem pored svih objekata ili selektujemo neki pojedinačni objekat.

3.6.3.7 Komanda Combine

Na prvi pogled ova komanda je ista kao i komanda **Group**, međutim postoje bitne razlike. Upotreba **Combine** je ista kao i **Group**. Komandu aktiviramo nakon što smo selektovali objekte koje želimo kombinovati. Komanda **Combine** predstavlja specijalni oblik komande **Group** pomoću koje *Corel* pamti odabrane objekte. Ili nije na način kojim manje angažuje radne resurse vašeg računara, čime omogućuje da formirate složenije crteže. Postoji više razloga za korišćenje ove komande, a to su bolje iskorišćenje memorije, mogućnost da radite sa čvorovima više objekata, što nije moguće kod komande **Group**, mogućnost da spojite više međusobno odvojenih linija, da poravnavate čvorove više nerevnih objekata. Važnost ove komande ćete veri vatno više osetiti kada steknete malo prakse i kada vaši crteži budu složeniji. Vrlo zanimljive efekte ćete postići primenom ove komande na obojene objekte.

3.6.3.8 Komanda Break Apart

Ova komanda je suprotnog dejstva komandi **Combine**. Omogućiti da pretvorite jedan objekat koji se sastoji od više zatvorenih linija u više objekata sastavljenih od pojedinačnih zatvorenih linija. Komanda se jednostavno aktivira izborom iz menija **Arrange** pošto ste selektovali željeni objekat. Primenom ove komande na tekstualni objekat ime sa rezultat razdvajanje svakog slova na posebnu zatvorenu krivu čiji oblik možete menjati uz pomoć čvorova.

3.6.3.9 Komanda Separate

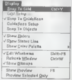
Ova komanda je nova u odnosu na prethodne verzije *Corela* i omogućuje da izdvojite originalni objekat nakon izvršenja komandi **Blend** i **Extrude** ili da izdvojite tekst koji je porevnat na liniju komandom **Fit Text To Path**.

3.6.3.10 Komanda Convert To Curves

Ova komanda pretvara odabrani pravougaonik, elipsu ili tekstualni objekat u objekat sačinjen od linija. U delu u kojem je bila reč o radu sa malom strelicom, pomenuto je uređenje u radu sa pravougaonikom i elipsom. Kada na željenom objektu aktivirate komandu **Convert to Curves** biće vam omogućen rad sa malom strelicom i na ovim objektima. Ako ovu komandu aktivirate na tekstualnom objektu, ona će pretvoriti vaš tekst u jedan objekat sačinjen od više zatvorenih linija. Ovde treba uočiti razliku u odnosu na komandu **Break Apart** koja bi odvojila svako slovo vašeg teksta kao posebni objekat. Upotrebom komande **Convert to curves**, dobijate najviše slobode da određeni tekst pretvorite u svoj logotip.

3.7 Meni Display

Komande ovog menija (slika 3.50) uglavnom su orijentisane na definisanje prametara izgleda prozora za crtanje.



Slika 3.50

3.7.1 Komanda Snap To Grid

Ova komanda omogućuje da svi pomaci i dužine predmeta u *Corelu* budu vezani za mrežu. **Grid** odnosno mreža čine tačkice koje prekrivaju vaš radni prostor i predstavljaju olakšicu za crtanje paralelnih i normalnih linija, pomak i ravnanje objekata. Mreža se postavlja i njeni parametri podešavaju komandom **Grid Setup**. Ako je ova komanda aktivirana, moći ćemo lakše da crtamo objekte kod kojih se zahteva velika preciznost. Svi pomaci će biti definisani sa celim brojem grid razmaka (razmaka između dve tačke mreže).

3.7.2 Komanda Grid Setup

Aktiviranjem ove komande pojaviće se **Grid Setup** dijalog-boks (slika 3.51) koji je podeljen u više celina. **Grid origin** predstavlja celinu kojom je uz pomoć vrednosti **Horizontal** i **Vertical** moguće definisati koordinantni početak mreže u odnosu na donji levi ivicu papira na kojem crtate. Pozitivna vrednosti će početak mreže pomerati gore desno, a negativna dole levo kako se vidi iz primera koordinantnog početka prikazanog u dijalog boksu. **Grid Frequency** definiše gustoću mreže – uz pomoć parametara **Vertical** i **Horizontal**, moći ćete da definišete gustoću mreže i to u broju tačaka po odabranoj jedinici mere. Za milimetre ovaj broj može da ima vrednosti između 0.02 i 2.83. (Na primer, za mrežu sa međusobnim razmakom tačkica od 0.5 mm, vrednost **Grid** frekvencije mora da bude H – 2 i V – 2.) Ukoliko ne zadate ispravne vrednosti, program će vas na to upozoriti. Ovaj dijalog boks omogućuje da definišete da li će se mreža videti ili ne (sa komandom **Show grid**) te da definišete i iz ovog dijalog boksa **Snap To Grid**.